Curso 2013 - 2014

Grado en Ingeniería Informática

TFG

Trabajo de Fin de Grado: Oracle Business Intelligence for the Enterprise

Héctor Lee Marco | NIA: 100074110

Index

1. Introduction

Business intelligence (BI) is a set of theories, methodologies, architectures, and technologies that transform raw data into meaningful and useful information for business purposes. Most of the companies generate a huge amount of data from transactions, logs, receipts and so on. Normally they only use it for administrative and legal reasons, but there is a great potential for this apparently useless information. BI can handle this enormous amount of unstructured data to help identify, develop and otherwise create new opportunities. BI, in simple words, makes interpreting voluminous data friendly. Making use of new opportunities and implementing an effective strategy can provide a competitive market advantage and long-term stability.

1.1. Motivation:

This project is going to create a study about the practical application of the BI tool created by oracle called OBIEE, Oracle Business Intelligence Enterprise Edition.

The aim of this project is to expose and explain oracle technologies in BI throughout detailed research, documentation and proof of concept, give an example of practical use in big enterprises, how it works and how it can benefit and make a difference in big companies. At the same time it is intended to make known and promote OBIEE as well as IT in the academic and professional environment.

For a better understanding, it is highly advisable to have an average understanding of databases, SQL, IT, statistics and a little bit of business administration and economy. This is due to the fact that this is computer engineering applied to the business in order to increase a company income or to optimize certain processes efficiency.

Consequently, the technologies used in this project will be detailed and justified.

1.1.1. OBIEE

As stated before, OBIEE is a Business Intelligence tool created by Oracle: Oracle Business Intelligence Enterprise Edition. OBIEE makes corporate data easier for business users to access, provides a common infrastructure for producing and delivering enterprise reports, scorecards, dashboards, ad-hoc analysis, and OLAP analysis

OBIEE uses data from the Enterprise to facilitate decision making. It helps the understanding of how a company works and functions, anticipates events with the objective of backing an enterprise decision.

1.1.2. Prove of Concept, Financial reporting in an OBIEE Dashboard

After the explanation of all the technologies that will be used, there will be a prove of concept that will create a web application in OBIEE with financial reports of an international company presented in an OBIEE Dashboard.

In the Prove of Concept section, first of all I will detail all the requirements needed and then proceed to develop the OBIEE web application and explain the followed steps.

Secondly, using the Oracle BI Administration Tool I will do all the data modelling and explain how the tool works and how the data modelling is done to prepare an RPD which is needed to create reports or analysis. The RPD is a repository file that contains all the metadata of the web application (more precisely, of the BI server). Then all the data that will be used will be modelled here.

After we have RPD and the data modelled, this RPD file will be uploaded to OBIEE as a new Subject Area. Using Answers (an Oracle BI component), new reports (also referred to as analysis) will be created using this Subject Area to meet the needs of the requirements and there will be an explanation of how an average user can easily create custom analysis and reports.

Then I will include these new created reports in dashboards that summarize the most important information and are fully prepared to help to make important big scale decisions. This last process will also be described step by step. And finally we will see the fully created Dashboard that shows the potential of OBIEE.

1.2. Project Target

The main target of this project is to research, explain and promote the usage and OBIEE and let people know about its benefits. An OBIEE web application will be created as a proof of concept that will unify all the researched technologies and the components created in this TFG.

This project will be divided in the following phases:

- Research, analysis and understanding of the present BI technologies that will be used in this project:
    - Oracle Database
    - Oracle Grid Architecture
    - Oracle Data Warehousing
    - OBIEE
- Detailed explanation of OBIEE and its arquitecture. First of all the concept of Service Oriented Arquitecture which OBIEE uses will be thoroughly explained. Then OBIEE High Level Architecture will be presented and all its components explained one by one. And finally OBIEE running architecture and how OBIEE works whenever a user requests any report, analysis or Dashboard.
- A proof of concept that will apply and unify everything that has been previously researched and detailed in this project and will show OBIEE in action. In this section, there will be a requirements specification which will be fulfilled in the rest of the proof of concept: data modelling, design of analysis or report creation and dashboard creation.

1.3. Document structure

- Section 1, introduction: this section gives a general idea about the project, its motivation, the main target that I want to achieve and the resources used.
- Section 2, state of art: Here I explain and give a general vision of BI technologies, Oracle and Oracle BI.
- Section 3, architecture: In here I talk about OBIEE architecture in depth: relevant concepts, it´s architecture and components and how it works.
- Section 4, proof of concept: The application created will be explained. Its requirements and how those requirements are met.
- Section 5, future projects: Suggestions from the student for possible future projects or extensions of the present project.
- Section 6, conclusions: Here I will expose my conclusions after the completion of this project.
- Section 7, bibliography: A listing of all the books, documents, articles, webpages that have been consulted for this project.
- Section 8, Appendix: In this section the Budget estimation, Task Planning and the Regulation Framework for this project will be detailed.

1.4. Resources

In this section the required resources to develop this project and run OBIEE and the application will be listed.

1.4.1. Requirements for the installation of OBIEE

OBIEE is part of Oracle Fusion Middleware and its requirements can be found in the Oracle document "Oracle Fusion Middleware, System Requirements and Specifications for Oracle Forms and Reports 11g Release". Here are the must have components for OBIEE installation.

- General Memory Requirements:
    - Operating System: Windows, UNIX or Linux.
    - Minimum Physical Requirements: 2 GB
    - Minimum Available Memory Required: 4 GB
- Memory and Disk Space Requirements for Installation and Configuration
    - Physical Memory: 922 MB
    - Temp Space: greater than 270 MB
    - Swap Space: greater than 500 MB
    - Minimum Available Memory Required: greater than 1422 MB
    - Disk Space: 3600MB for installation and 810 MB for configuration.
- Startup Requirements:
    - CPU Speed: at least 300 MHZ
    - Temp Space: at least 270 MB
    - Swap Space: at least 500 MB
    - Monitor: at least 256 colors
- The machine from where OBIEE is accessed must have access through internet or any network to where Oracle BI Domain is installed.

1.4.2. Other Requirements

A machine from where the user accesses OBIEE. This machine must have the minimum requirements to efficiently run any of the following browsers: IE, Google Chrome or Firefox.

1.5. Glossary of terms

- OBIEE: Oracle Business Enterprise Edition.
- BI: Business intelligence.
- SQL: Structured Query Language.
- IT: Information Technology.
- Ad-hoc query: is the dynamic creation of queries. When the queries can't be priory created, they are dynamically created on their execution and this is called Ad-hoc query.
- RPD: is a repository file that contains all the metadata of the web application (more precisely, of the BI server).
- Answers: is the OBIEE component where analysis and reports are created.
- Dashboard: is the combination of prompts, reports, graphics and other elements in pages or tabs prepared to summarize important information for the Enterprise decision making.
- SAO: Service Oriented Arquitecture. More details in section 3.1.
- DBMS: Database Management System is a suite of computer software providing the interface between users and a database or databases.
- Data warehouse: a relational database that is designed for query and analysis rather than for transaction processing.
- OLTP system: online transaction processing (OLTP) system.
- ELT: Extract Load and Transform.
- CRM: Customer relationship management is a system for managing a company's interactions with current and future customers.
- HR: Human Resources.
- XML: Extensible Markup Language.
- WSDL: Web Services Description Language
- JMX: Java Management Extensions
- BI Publisher: Is an Oracle BI Domain component that provides an enterprise reporting solution for authoring, managing, and delivering all types of highly formatted documents to employees, customers, and suppliers.

- Oracle RTD: Real-Time Decisions, this is also an Oracle Domain component that provides enterprise analytics software solutions that enable companies to make better decisions in real-time at key, high-value points in operational business processes.
- KPI: Key Performer Indicator.
- ODBC: Oracle Database Connection.
- OPMN: Oracle Process Manager and Notification Server.
- MUD: Multi-User-Development.
- LTS: Logical Table Sources.
- Session Variables: variables that are unique to each user's session.
- Repository Variables: global variables where all users see the same value.
- Presentation Variables: are variables that can be created by report developers and be referenced in their reports. Users are able to change the value assigned to a Presentation Variable using a "Dashboard Prompt".

2. State of the Art

BI technologies provide historical, current and predictive views of business operations. Common functions of business intelligence technologies are reporting, online analytical processing, analytics, data mining, process mining, complex event processing, business performance management, benchmarking, text mining, predictive, descriptive and prescriptive analytics.

This section will give a global vision about the main technologies used in this project.

There will be a brief explanation about its history and evolution throughout its life, the main characteristics and other relevant information.

2.1. Databases

A database is an organized collection of data. The data are typically organized to model relevant aspects of reality in a way that supports processes requiring this information.

Formally, "database" refers to the data themselves and supporting data structures. Databases are created to operate large quantities of information by inputting, storing, retrieving and managing that information. Databases are set up so that one set of software programs provides all users with access to all the data.

A "database management system" (DBMS) is a suite of computer software providing the interface between users and a database or databases. Because they are so closely related, the term "database" when used casually often refers to both a DBMS and the data it manipulates.

The interactions catered for by most existing DBMSs are:

- Data definition: Defining new data structures for a database, removing data structures from the database, modifying the structure of existing data.
- Update: Inserting, modifying, and deleting data.
- Retrieval: Obtaining information either for end-user queries and reports or for processing by applications.

Administration: Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information if the system fails.

A DBMS is responsible for maintaining the integrity and security of stored data, and for recovering information if the system fails.

2.2. Oracle

Oracle Corporation is a U.S. based multinational computer technology corporation headquartered in Redwood City, California, United States. The company specializes in developing and marketing computer hardware systems and enterprise software products – particularly its own brands of database management systems. Oracle is the second-largest software maker by revenue, after Microsoft.

2.2.1. History

Oracle Corporation, formerly Software Development Laboratories (1977–79), Relational Software Inc. (1979–82), and Oracle Systems Corporation (1982–95), global corporation that develops and markets computer software applications for business. The company is best known for its Oracle database software, a relational database management system, and for computer systems and software, such as Solaris and Java, acquired in its purchase of Sun Microsystems in 2010. Oracle is based in Redwood Shores, California.

The company, initially called Software Development Laboratories, was founded in 1977 by Lawrence (Larry) Ellison and Robert (Bob) Miner, computer programmers at the American electronics company Ampex Corporation, and by Edward (Ed) Oates, Ellison's supervisor at Ampex. Inspired by a research paper written by British-born computer scientist Edgar F. Codd that outlined a relational database model, Ellison and his colleagues saw commercial potential in the approach, which organized large amounts of data in a way that allowed for efficient

storage and quick retrieval. The trio set to work developing and marketing a program based on Codd's data management theory. In 1979 the company released Oracle, the earliest commercial relational database program to use Structured Query Language (SQL), and the versatile database program quickly became popular. Its first customer was the U.S. Air Force, which used the program at Wright-Patterson Air Force Base, near Dayton, Ohio. Part of Oracle Corporation's early success arose from using the C programming language to implement its products. This eased porting to different operating systems (most of which support C).

Known for innovation and aggressive marketing, the company, renamed Oracle in 1982 after its flagship product, grew rapidly throughout the 1980s, going public in 1986. In 1987 Oracle became the largest database management company in the world. Although Oracle's eponymous database has seen steady growth, much of Oracle's growth has come through its aggressive acquisitions of software companies with products for a range of business and technology applications. In its history Oracle lays claim to buying more than 50 companies, including high-profile multibillion-dollar purchases of PeopleSoft, Siebel, BEA, and Sun Microsystems.

2.2.2. Oracle database

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

2.2.3. Oracle Grid Architecture

Grid computing is a new IT architecture that produces more resilient and lower cost enterprise information systems. With grid computing, groups of independent, modular hardware and software components can be connected and rejoined on demand to meet the changing needs of businesses.

The grid style of computing aims to solve some common problems with enterprise IT: the problem of application silos that lead to under-utilized, dedicated hardware resources, the problem of monolithic, unwieldy systems that are expensive to maintain and difficult to change, and the problem of fragmented and disintegrated information that cannot be fully exploited by the enterprise as a whole.

Benefits of Grid Computing Compared to other models of computing, IT systems designed and implemented in the grid style deliver higher quality of service, lower cost, and greater flexibility. Higher quality of service results from having no single point of failure, a robust security infrastructure, and centralized, policy-driven management. Lower costs derive from increasing the utilization of resources and dramatically reducing management and maintenance costs. Rather than dedicating a stack of software and hardware to a specific task, all resources are pooled and allocated on demand, thus eliminating under-utilized capacity and redundant capabilities. Grid computing also enables the use of smaller individual hardware components, thus reducing the cost of each individual component and providing more flexibility to devote resources in accordance with changing needs.

The grid style of computing treats collections of similar IT resources holistically as a single pool, while exploiting the distinct nature of individual resources within the pool. To address simultaneously the problems of monolithic systems and fragmented resources, grid computing achieves a balance between the benefits of holistic resource management and flexible independent resource control. IT resources managed in a grid include:

- Infrastructure: the hardware and software that create a data storage and program execution environment
- Applications: the program logic and flow that define specific business processes
- Information: the meanings inherent in all different types of data used to conduct business

Core Tenets of Grid Computing Two core tenets uniquely distinguish grid computing from other styles of computing, such as mainframe, client-server, or multi-tier: virtualization and provisioning.

- With virtualization, individual resources (e.g. computers, disks, application components and information sources) are pooled together by type then made available to consumers (e.g. people or software programs) through an abstraction.

Virtualization means breaking hard-coded connections between providers and consumers of resources, and preparing a resource to serve a particular need without the consumer caring how that is accomplished.

- With provisioning, when consumers request resources through a virtualization layer, behind the scenes a specific resource is identified to fulfill the request and then it is allocated to the consumer. Provisioning as part of grid computing means that the system determines how to meet the specific need of the consumer, while optimizing operation of the system as a whole.

The specific ways in which information, application or infrastructure resources are virtualized and provisioned are specific to the type of resource, but the concepts apply universally. Similarly, the specific benefits derived from grid computing are particular to each type of resource, but all share the characteristics of better quality, lower costs and increased flexibility.

Infrastructure Grid Infrastructure grid resources include hardware resources such as storage, processors, memory, and networks as well as software designed to manage this hardware, such as databases, storage management, system management, application servers, and operating systems.

Virtualization and provisioning of infrastructure resources mean pooling resources together and allocating to the appropriate consumers based on policies. For example, one policy might be to dedicate enough processing power to a web server that it can always provide sub-second response time. That rule could be fulfilled in different ways by the provisioning software in order to balance the requests of all consumers.

Treating infrastructure resources as a single pool and allocating those resources on demand saves money by eliminating under-utilized capacity and redundant capabilities. Managing hardware and software resources holistically reduces the cost of labor and the opportunity for human error.

Spreading computing capacity among many different computers and spreading storage capacity across multiple disks and disk groups removes single points of failure so that if any individual component fails, the system as a whole remains available. Furthermore, grid computing affords the option to use smaller individual hardware components, such as blade servers and low cost storage, which enables incremental scaling and reduces the cost of each individual component, thereby giving companies more flexibility and lower cost.

Infrastructure is the dimension of grid computing that is most familiar and easy to understand, but the same concepts apply to applications and information.

Applications Grid Application resources in the grid are the encodings of business logic and process flow within application software. These may be packaged applications or custom

applications, written in any programming language, reflecting any level of complexity. For example, the software that takes an order from a customer and sends an acknowledgement, the process that prints payroll checks, and the logic that routes a particular customer call to a particular agent are all application resources.

Historically, application logic has been intertwined with user interface code, data management code, and process or page flow and has lacked well-defined interfaces, which has resulted in monolithic applications that are difficult to change and difficult to integrate.

Service oriented architecture has emerged as a superior model for building applications, and service oriented architecture concepts align exactly with the core tenets of grid computing. Virtualization and provisioning of application resources involves publishing application components as services for use by multiple consumers, which may be people or processes, then orchestrating those services into more powerful business flows.

In the same way that grid computing enables better reuse and more flexibility of IT infrastructure resources, grid computing also treats bits of application logic as a resource, and enables greater reuse of application functionality and more flexibility in changing and building new composite applications.

Furthermore, applications that are orchestrated from published services are able to view activities in a business as a single whole, so that processes are standardized across geography and business units and processes are automated end-to-end. This generates more reliable business processes and lowers cost through increased automation and reduced variability.

Information Grid The third dimension to grid computing, after infrastructure and applications, is information. Today, information tends to be fragmented across a company, making it difficult to see the business as a whole or answer basic questions about customers. Without information about who the customer is, and what they want to buy, information assets go underexploited.

In contrast, grid computing treats information holistically as a resource, similar to infrastructure and applications resources, and thus extracts more of its latent value. Information grid resources include all data in the enterprise and all metadata required to make that data meaningful. This data may be structured, semi-structured, or unstructured, stored in any location, such as databases, local file systems, or e-mail servers, and created by any application.

The core tenets of grid computing apply similarly to information as they do to infrastructure and applications. The infrastructure grid exploits the power of the network to allow multiple servers or storage devices to be combined toward a single task, then easily reconfigured as needs change. A service oriented architecture, or an applications grid, enables independently developed services, or application resources, to be combined into larger business processes, then adapted as needs change without breaking other parts of the composite application.

Similarly, the information grid provides a way for information resources to be joined with related information resources to greater exploit the value of the inherent relationships among information, then for new connections to be made as situations change.

The relational database, for example, was an early information virtualization technology. Unlike its predecessors, the network database and hierarchical database models, in which all relationships between data had to be predetermined, relational database enabled flexible access to a general-purpose information resource. Today, XML furthers information virtualization by providing a standard way to represent information along with metadata, which breaks the hard link between information and a specific application used to create and view that information.

Information provisioning technologies include message queuing, data propagation, replication, extract-transform-load, as well as mapping and cleansing tools to ensure data quality. Data hubs, in which a central operational data store continually syncs with multiple live data sources, are emerging as a preferred model for establishing a single source of truth while maintaining the flexibility of distributed control.

Grid Resources Work Well Independently and Best Together By managing any single IT resource – infrastructure, applications, or information - using grid computing, regardless of how the other resources are treated, enterprises can realize higher quality, more flexibility, and lower costs. For example, there is no need to rewrite applications to benefit from an infrastructure grid. It is also possible to deploy an applications grid, or a service oriented architecture, without changing the way information is managed or the way hardware is configured.

It is possible, however, to derive even greater benefit by using grid computing for all resources. For example, the applications grid becomes even more valuable when you can set policies regarding resource requirements at the level of individual services and have execution of different services in the same composite application handled differently by the infrastructure - something that can only be done by an application grid in combination with an infrastructure grid. In addition, building an information grid by integrating more information into a single source of truth becomes tenable only when the infrastructure is configured as a grid, so it can scale beyond the boundary of a single computer.

2.3. Oracle Data Warehousing

A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but can include data from other sources. Data warehouses separate analysis workload from transaction work load and enable an organization to consolidate data from several sources. This helps in:

- Maintaining historical records
- Analyzing the data to gain a better understanding of the business and to improve the business.

In addition to a relational database, a data warehouse environment can include an extraction, transportation, transformation , and loading (ETL) solution, statistical analysis, reporting, data mining capabilities, client analysis tools, and other applications that manage the process of gathering data, transforming it into useful, actionable information, and delivering it to business users.

2.3.1. Subject Oriented

Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a data warehouse that concentrates on sales. Using this data warehouse, you can answer questions such as "Who was our best customer for this item last year?" or "Who is likely to be our best customer next year?" This ability to define a data warehouse by subject matter, sales in this case, makes the data warehouse subject oriented.

2.3.2. Integrated

Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as naming conflicts and inconsistencies among units of measure. When they achieve this, they are said to be integrated.

2.3.3. Nonvolatile

Nonvolatile means that, once entered into the data warehouse, data should not change. This is logical because the purpose of a data warehouse is to enable you to analyze what has occurred.

2.3.4. Time Variant

A data warehouse's focus on change over time is what is meant by the term time variant. In order to discover trends and identify hidden patterns and relationships in business, analysts need large amounts of data. This is very much in contrast to online transaction processing (OLTP) systems, where performance requirements demand that historical data be moved to an archive.

2.3.5. Contrasting OLTP and Data Warehousing

OLTP		Data Warehouse
Complex data structures (3NF databases)		Multidimensional data structures
Few	**Indexes**	Many
Many	**Joins**	Some
Normalized DBMS	**Duplicated Data**	Denormalized DBMS
Rare	**Derived Data and Aggregates**	Common

One major difference between the types of system is that data warehouses are not usually in third normal form (3NF), a type of data normalization common in OLTP environments.

Data warehouses and OLTP systems have very different requirements. Here are some examples of differences between typical data warehouses and OLTP systems:

2.3.5.1. Workload

Data warehouses are designed to accommodate ad hoc queries and data analysis. You might not know the workload of your data warehouse in advance, so a data warehouse should be optimized to perform well for a wide variety of possible query and analytical operations.

OLTP systems support only predefined operations. Your applications might be specifically tuned or designed to support only these operations.

2.3.5.2. Data modifications

A data warehouse is updated on a regular basis by the ETL process (run nightly or weekly) using bulk data modification techniques. The end users of a data warehouse do not directly update the data warehouse except when using analytical tools, such as data mining, to make predictions with associated probabilities, assign customers to market segments, and develop customer profiles. In OLTP systems, end users routinely issue individual data modification

statements to the database. The OLTP database is always up to date, and reflects the current state of each business transaction.

2.3.5.3. Schema design

Data warehouses often use denormalized or partially denormalized schemas (such as a star schema) to optimize query and analytical performance. OLTP systems often use fully normalized schemas to optimize update/insert/delete performance, and to guarantee data consistency.

2.3.5.4. Typical operations

A typical data warehouse query scans thousands or millions of rows. For example, "Find the total sales for all customers last month". A typical OLTP operation accesses only a handful of records. For example, "Retrieve the current order for this customer."

2.3.5.5. Historical data

Data warehouses usually store many months or years of data. This is to support historical analysis and reporting. OLTP systems usually store data from only a few weeks or months. The OLTP system stores only historical data as needed to successfully meet the requirements of the current transaction.

2.4. OBIEE

OBIEE is based in an Information System that uses data extracted from production data and information related to the enterprise, the business and other economic data.

With ELT (Extract Load and Transform like ODI: Orcale Data Integrator) or ETL (Extract Transfer and Load, like Informatica) tools and techniques data from different sources is extracted and prepared to be loaded in a data warehouse where OBIEE can make use of it to create a fast and efficient Analytical and reporting system.

OBIEE features:

Interactive Dashboards which provides fully interactive dashboards and reports with a rich variety of visualizations

Self-serve Interactive Reporting that enables business users to create new analyses from scratch or modify existing analyses without any help from a specialist.

Enterprise Reporting allowing the creation of highly formatted templates, reports, and documents such as flash reports, checks, and more.

Proactive Detection and Alerts also provides a powerful, near-real- time, multi-step alert engine that can trigger workflows based on business events and notify stakeholders via their preferred medium and channel.

Actionable Intelligence that turns insights into actions by providing the ability to invoke business processes from within the business intelligence dashboards and reports.

Microsoft Office Integration enables users to embed up-to-the-minute corporate data in Microsoft PowerPoint, Word, and Excel documents.

Spatial Intelligence via Map-based Visualizations helps users to visualize their analytics data using maps, bringing the intuitiveness of spatial visualizations to the world of business intelligence.

2.4.1. OBIEE's History

The current version of OBIEE is 11g and immediate previous version is 10.1.3. The base product of OBIEE is Siebel Analytics which was a product of Siebel Corporation and later this was taken over by Oracle Corporation and re-named as OBIEE. Prior to Siebel Analytics this product was originally built by a company called nQuire and it used to be referred as nQuire software in older days.

Siebel is one of the market leaders in CRM application and owning few of domain specific Applications like CRM, Finance, Retail, etc.

Siebel Analytical software had two lines of products:

- Siebel Analytics
- Siebel Application

Siebel Analytics worked as a standalone analytical application which provided data analytical capabilities on any type of physical data base. For this we needed to build a standalone repository.

Siebel Applications were prebuilt components/products which build based on real world functional domain like Sales, Finance, HR, Pharma etc. When a customer bought this application he would get a set of pre-built repositories/reports/dashboards specific to each functional application. This would build a data warehouse from existing OLTP application systems using Informatica pre-built mappings and DAC as a scheduling software.

In 2006/2007 oracle was looking for a best analytical tool for its customers especially for data warehousing reporting purpose. It worked out a deal to buy entire Siebel Corp, as a result Siebel analytics became an Oracle product. This product was renamed to OBIEE and released as a full-fledged reporting and other Applications were renamed to BI Apps.

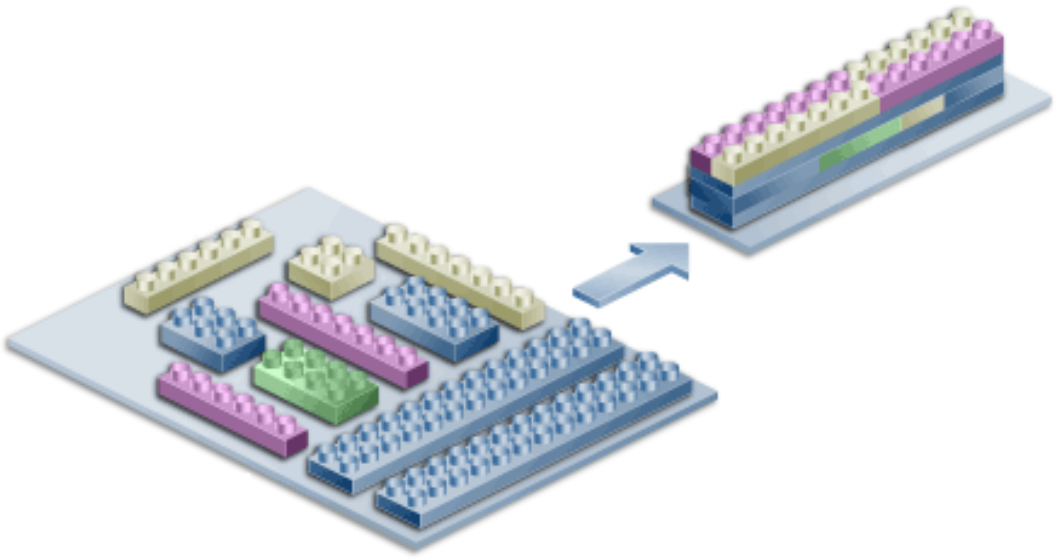
Last version of Siebel analytics was 7.7.

3. Arquitecture

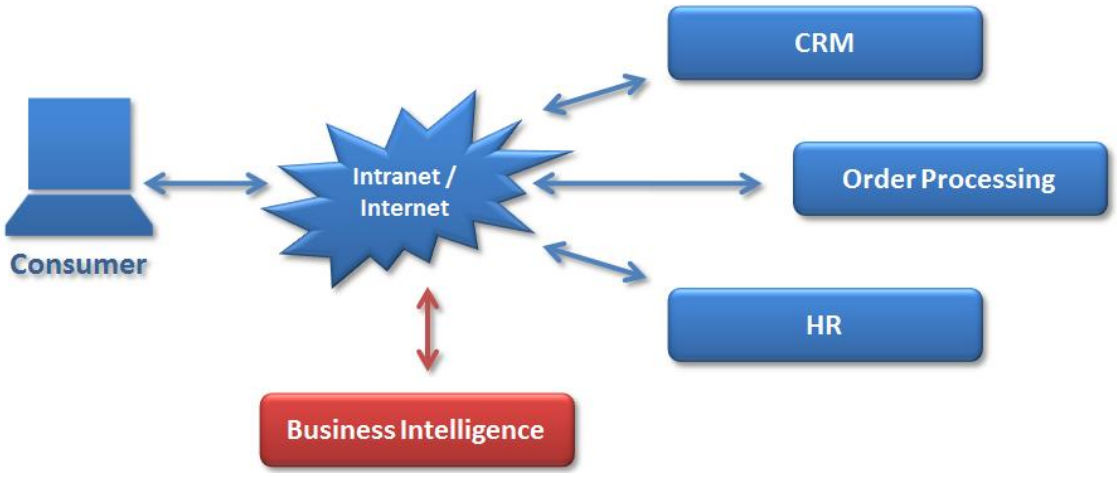
In this section I will give a detailed explanation of OBIEE and its arquitecture. First of all the concept of Service Oriented Arquitecture which OBIEE uses will be thoroughly explained. Then OBIEE High Level Architecture will be presented and all its components explained one by one. Finally OBIEE running architecture and how OBIEE works whenever a user requests any report, analysis or Dashboard will be explained as well.

3.1. Service Oriented Architecture

A Service Oriented Architecture (SOA) is an enterprise architecture consisting of modular web based services that can be easily integrated and reused, creating a truly flexible and adaptable IT infrastructure. Each service serves as a building block forming an architecture that supports multiple connected enterprise applications working together to provide streamlined solutions to business problems.

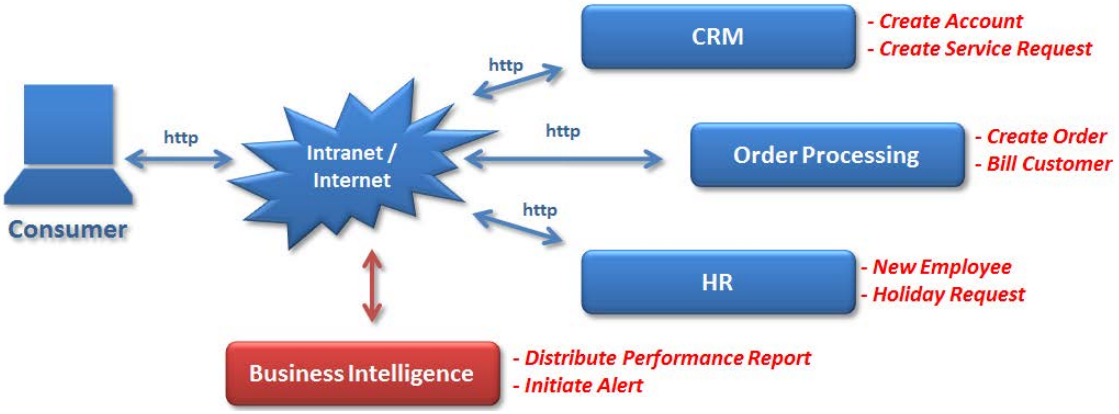


Behind the scenes, a SOA implementation can consist of a combination of technologies, products and support infrastructure elements. However, the key factor is that they all integrate via a common set of standards, how each building block is implemented at the back end is irrelevant.



In a SOA implementation typically:

- Communication will be performed over HTTP / HTTPS.
- Messages are delivered in XML format.
- Business functions/processes are presented as **Web Services**.



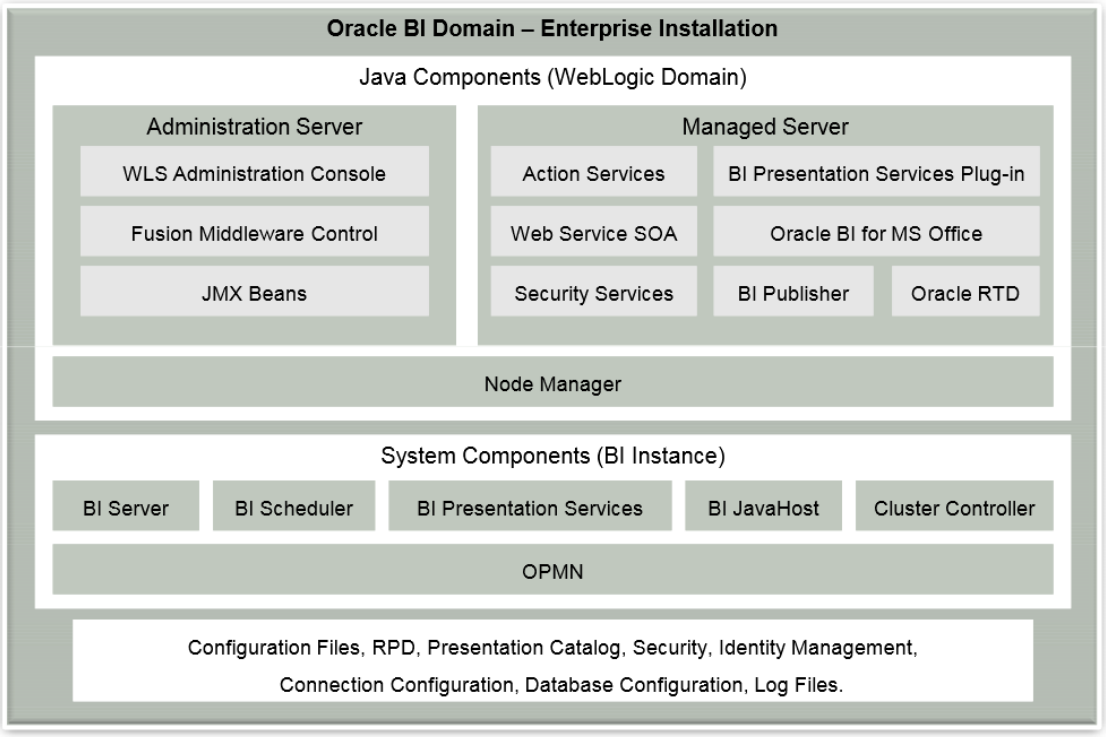
3.1.1. Web Services

Web services are programs that can be access remotely using XML-based languages. What each program can do is described in a standard XML format called Web Services Description Language (WSDL). The consumer does not need to know how the program is implemented and is only interested in what the program can do (as defined in the WSDL).

The Consumer sends a request in the form of a Simple Object Access Protocol (SOAP) message (SOAP is an XML messaging framework designed to allow heterogeneous applications to exchange structured information). The web service provider processes the request and returns the response in XML format. The Web service provider may require some form of credentials to be passed across, messaging may be encrypted.

There are several products that create web services, one of them is Oracle Jdeveloper.

3.2. OBIEE High Level Arquitecture



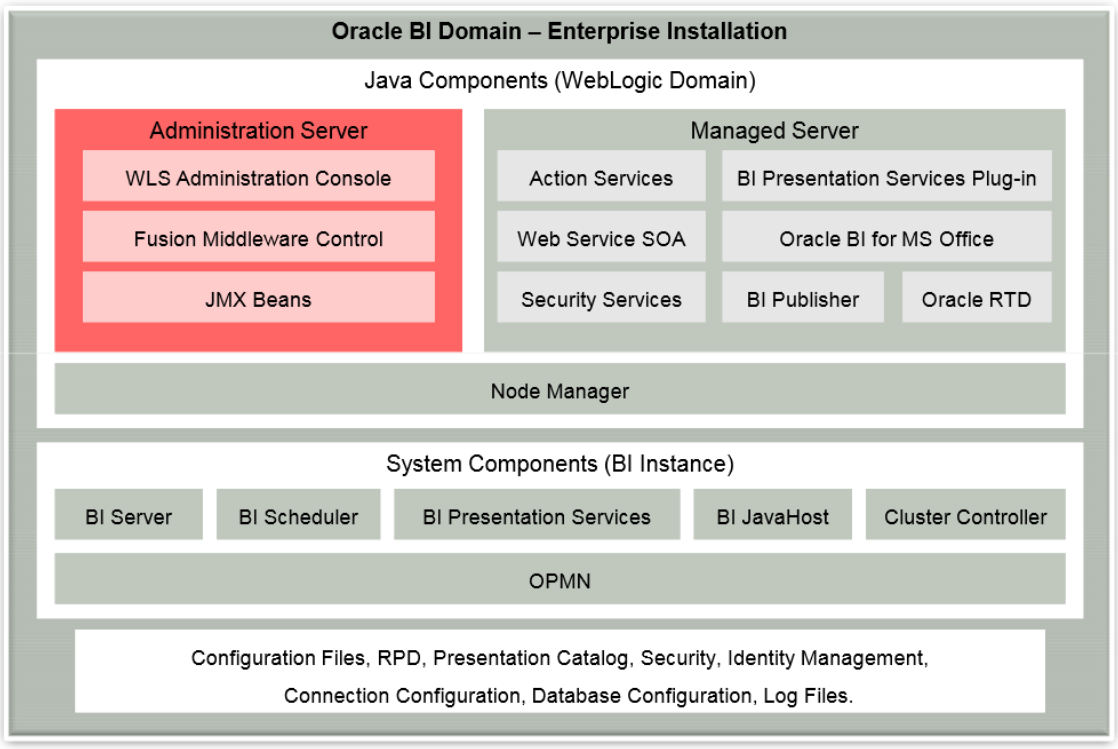
The entire system architecture is called BI Domain, this BI Domain divided into Java components and non-Java components. Java components are weblogic server Domain components and non-java components are Oracle BI system components.

3.2.1. Java Components (Weblogic Server Domain)

This domain consists of Managed server, Admin Server and node manager. These services comprises mainly with all the java modules to trigger the java services.

3.2.1.1. Administration Server

A JEE container that runs in a dedicated Java virtual machine that contains Java components for administering the system .It typically trigger the start, stop kind of admin activity for his peer Manager server processes.



3.2.1.1.1. WLS Administration Console

It is a Java EE application server that supports the deployment of Oracle Business Intelligence Java components. Oracle WebLogic Server Administration Console access has been provided by Fusion Middleware Control. Oracle WebLogic Server Administration Console enables to monitor and manage a WebLogic Server domain. Its capabilities include the following:

- Monitoring health and performance of JEE servers
- Configuring WebLogic domains
- Stopping and starting JEE servers
- Viewing JEE server logs

3.2.1.1.2. Fusion Middleware Control

Fusion Middleware Control is a browser-based tool and the recommended method for monitoring, managing, and configuring Oracle Business Intelligence components. It is also used for:

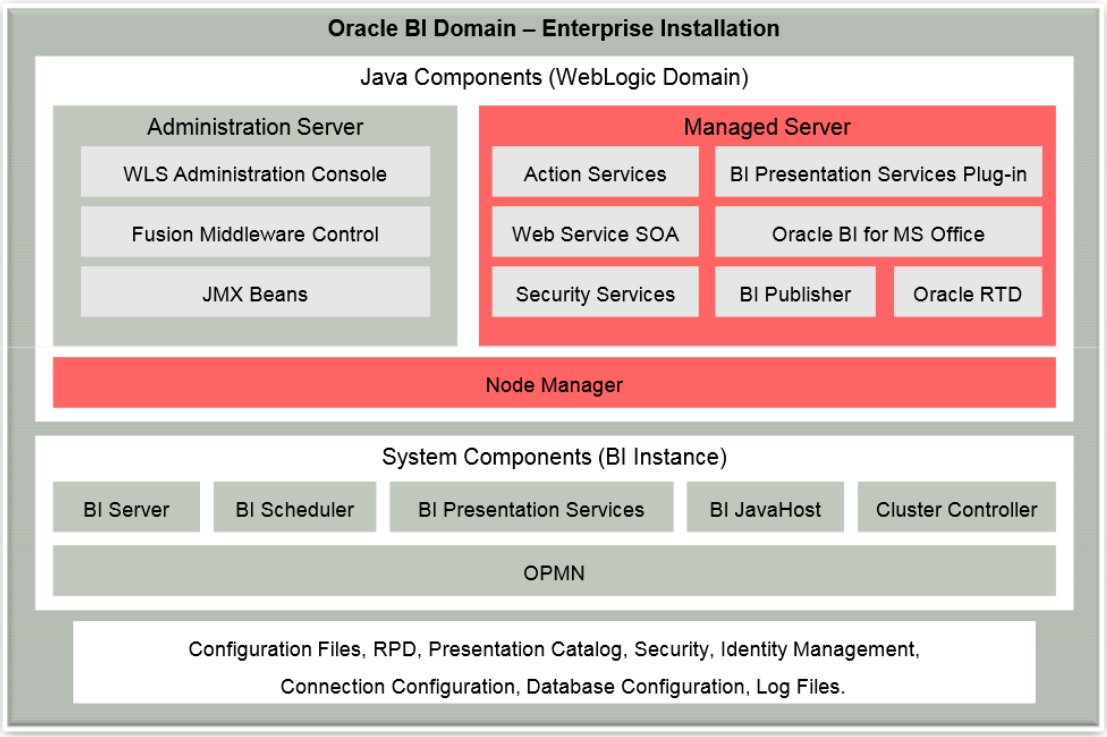
- Starting, stopping, and restarting all system components (BI Server, BI Presentation Server) and Managed Servers
- Configuring preferences and defaults
- Starting, stopping, and restarting all system components (BI Server,BI Presentation Server) and Managed Servers
- Managing performance and monitoring system metrics(DMS-Dynamic Monitoring System)
- Performing diagnostics and logging (ODL-Oracle Diagnostic Logging)Fusion Middleware Control also provides access to Oracle WebLogic Server Administration Console, where you monitor and manage Oracle Business Intelligence Java components.

3.2.1.1.3. Java components (JMX MBeans):

These are deployed as one or more Java EE applications. They are Administrative Components, Enterprise Management applications and JMX (Java Management Extensions) MBeans for managing all configuration and run-time settings for Oracle Business Intelligence.

3.2.1.2. Managed Server

A JEE container that runs in a dedicated Java virtual machine that provides the run-time environment for the Java-based services and applications within the system. The services comprises of BI plug-in, Security, publisher, SOA, BI Office services, etc.



3.2.1.2.1. Action Services

This component provides the dedicated Web services that are required by the Action Framework and that enable an administrator to manually configure which Web service directories can be browsed by users when they create actions.

3.2.1.2.2. BI Presentation Services Plug-in

A JEE application that routes HTTP and SOAP requests to Oracle BI Presentation Services.

3.2.1.2.3. Web Service SOA

This component provides dedicated Web services for objects in the Oracle BI Presentation Catalog, to invoke analyses, agents, and conditions. They make it easy to invoke Oracle Business Intelligence functionality from Business Process Execution Language (BPEL) processes.

3.2.1.2.4. Oracle BI for MS Office

This component provides the integration between Oracle Business Intelligence and Microsoft Office products.

3.2.1.2.5. Security Services

This component provides dedicated Web services that enable the integration of the Oracle BI Server with the Oracle Fusion Middleware security platform, for example, JPS (Java Platform Security), CSF (Credential Store Framework) and users and groups managed by BI LDAP security.

3.2.1.2.6. BI Publisher

This component provides an enterprise reporting solution for authoring, managing, and delivering all types of highly formatted documents to employees, customers, and suppliers.

3.2.1.2.7. Oracle RTD

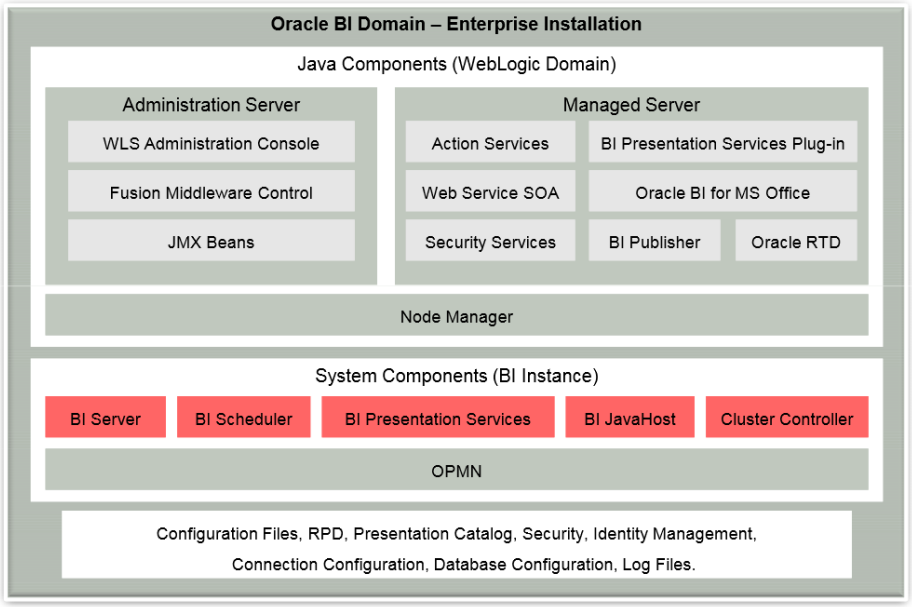
Oracle RTD, Real-Time Decisions, this component provides enterprise analytics software solutions that enable companies to make better decisions in real-time at key, high-value points in operational business processes.

3.2.1.3. Node Manager

Node Manager is a separate java utility runs to trigger the auto start, stop, restart activities and it provides process management services for the Admin server and Managed Server.

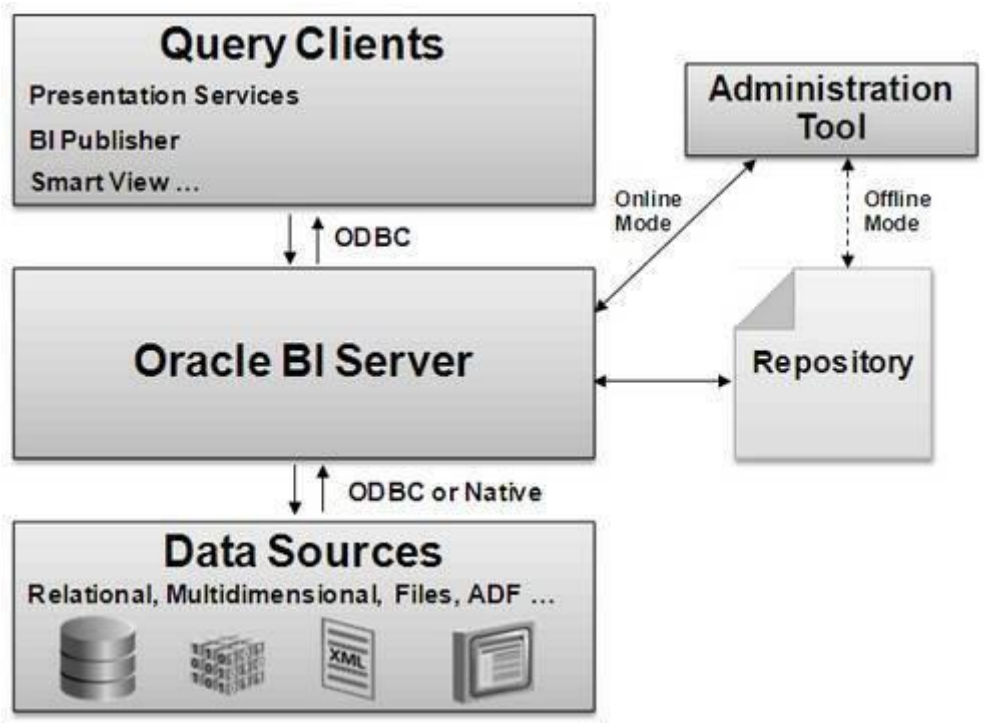
3.2.2. System components (BI Instance)

This system components are deployed as non-JEE components, such as processes and services written in C++ and J2SE.



3.2.2.1. BI Server

This component provides the query and data access capabilities at the heart of Oracle Business Intelligence and provides services for accessing and managing the enterprise semantic model (stored in a file with a .RPD extension).



The BI Server doesn't itself hold data, instead, it translates the incoming logical query into one or more outgoing" physical queries" against the relevant data sources. Clients of the Oracle BI Server see a logical schema view (the presentation layer) independent of the source physical database schemas.

Oracle BI Server clients (such as Presentation Service) submit simplified logical SQL, which ultimately gets translated by the server to some combination of physical SQL sent to the back-end databases (or files), in addition to intermediate processing within the Oracle BI Server Execution Engine.

At a simplified level, the internal layers provide a centralized and consistent conceptual data model and have two primary functions:

- Compile incoming query requests into executable code.
- Execute the code.

Repository design is the creation and configuration task of this three layer (presentation, logical and physical) and is the most important development part of BI Server. This will be explained with more detail in a following section.

The Oracle BI Server functions include:

- Session and query management, cancellation.
- Statistics logging (log).
- Monitoring (usage tracking, …).
- Security, using standard protocols, for example, LDAP, Active Directory or OID.

3.2.2.2. BI Scheduler

This component provides extensible scheduling for analyses to be delivered to users at specified times. (Oracle BI Publisher has its own scheduler)

3.2.2.3. BI Presentation Services

This component provides the framework and interface for the presentation of business intelligence data to Web clients. It maintains an Oracle BI Presentation Catalog service on the file system for the customization of this presentation framework.

3.2.2.4. BI JavaHost

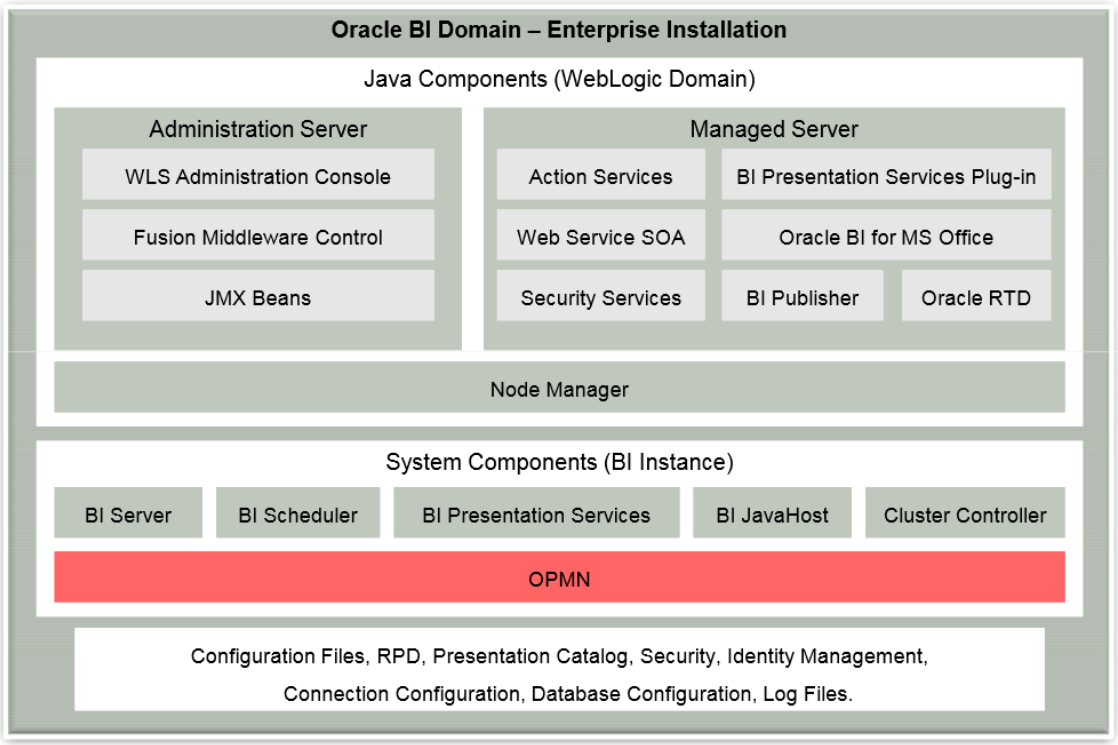
This component provides component services that enable Oracle BI Presentation Services to support various components such as Java tasks for Oracle BI Scheduler, Oracle BI Publisher, and graph generation.

3.2.2.5. Cluster Controller

This components distributes requests to the BI Server, ensuring requests are evenly load-balanced across all BI Server process instances in the BI domain.

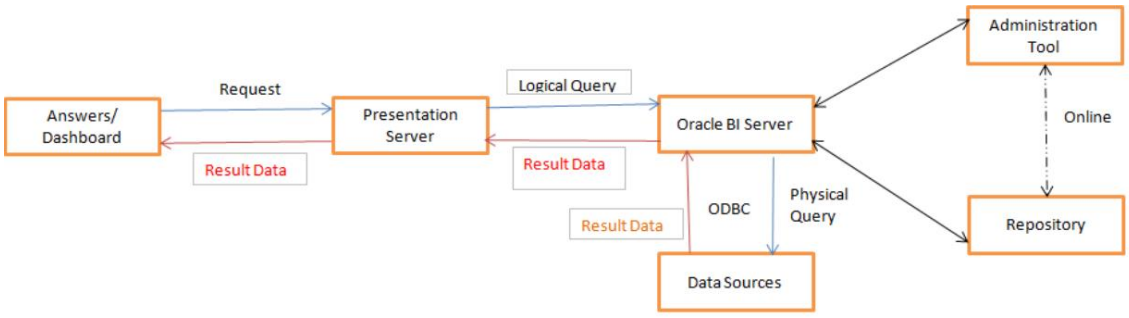
3.2.2.6. Oracle Process Manager and Notification Server (OPMN)

By using this OPMN services we can stop and start all system components of BI. It is monitored, managed and controlled by Fusion Middleware Controller.

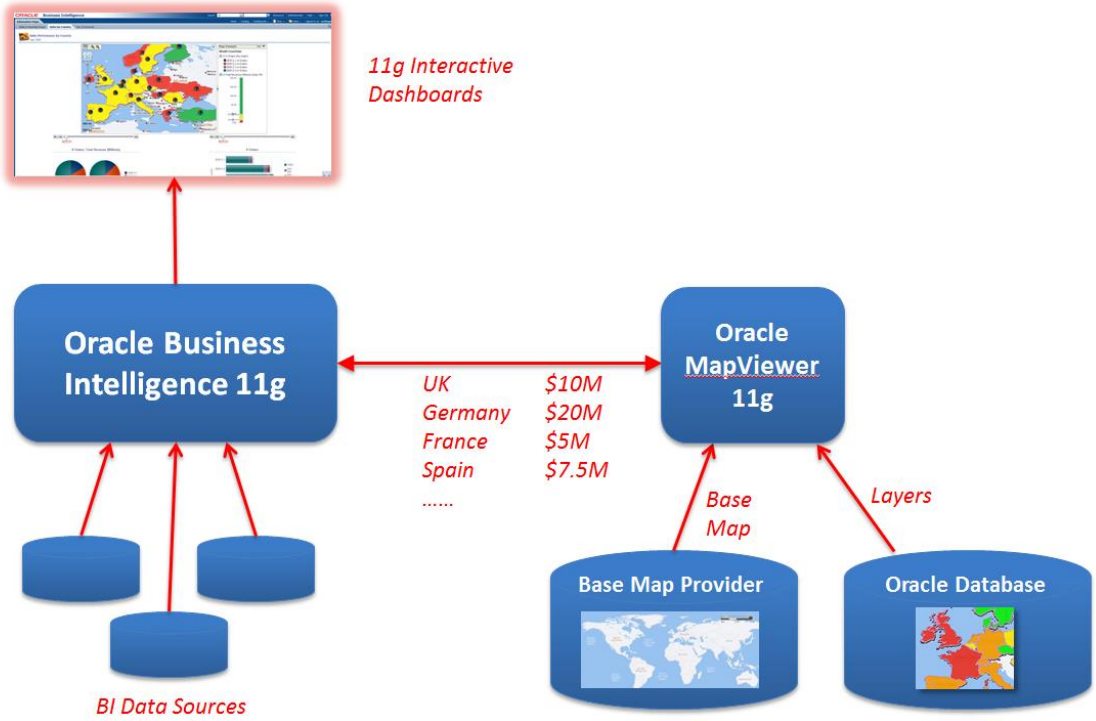


3.3. OBIEE Running Architecture

Whenever an OBIEE user runs a report in answers or a dashboard, OBIEE runs the following process. First of all, a request is made through Answers and sent to the Presentation server. The Presentation Server converts the request into logical SQL and sent to the BI server. The BI server converts logical SQL into physical SQL and then sends it to the database. The result from the database then gets back to the user through the same path.



In the case that there are embedded maps involved in the request, Oracle Map Viewer also intervenes in the process.



4. Proof of concept

In this section I will first define the requirements that the application created in OBIEE needs to accomplish. Then I will explain how the application has been created, modelled and designed, show the application and show how it runs to meet all the requirements.

4.1. Requirements Specification

4.1.1. Functional Requirements

These are the requirements that specify what the application or system must do. In other words, here is where the system or application functions are described.

The list of Functional requirements is as follows:

- FR-1: The application interface should be simple, easy to use and user friendly. The navigation between different sections, reports and dashboards should be easy and intuitive.
- FR-2: The user will be able to use the application in any of the following browsers: Firefox, IE or Chrome. No new windows or tabs will be created by the application unless the user explicitly asks for it.
- FR-3: The application will use English as standard language but will accept input from the user in other languages.
- FR-4: The application will let the user obtain crossed information or query (ad-hoc query included) related to Sales Orders:
    - Time
        - Date
        - Week
        - Month
        - Quarter
        - Year
    - Customer
        - Customer Account Number
        - Customer Type
        - Sales Chanel Code

        - Customer Class Code
    - Contact
        - Person Id
        - Email Address
        - Title
        - Full Name
        - Sex
        - Nationality
        - Effective Start Date
        - Effective End Date
        - Person Type
    - Organization
        - Organization Id
        - Organization Name
        - Business Group
    - Product
        - Product Id
        - Product Name
        - Product Type
    - Sales Representative
        - Sales Representative Id
        - Email Address
        - Title
        - Full Name
        - Sex
        - Nationality
        - Effective Start Date

        - Effective End Date
        - Person Type
    - Order
    - Number of Orders
    - Number of Order Lines
    - Number of Returns
    - Unit Cost
    - List Prize
    - Quantity
    - Order Amount
    - Return Amount
    - Cost of Sales
    - Total Revenue in millions
    - Sales Margin percentage
    - Returns per one thousand orders

- FR-5: The information described in FR-4 can be increase or decrease as per the client needs and the application must not suffer any drawbacks for this change.
- FR-6: Users must input a valid user name and password in order to access the application.
- FR-7: The application must have already developed reports and dashboards, but must let the user the possibility to develop basic reports and dashboards by himself.
- FR-8: The application must allow the user to sort selected information, ascending or descending. In addition it must let the users multiple sorting.
- FR-9: The application must let the user filter the results by any of the columns or elements mentioned in FR-4.
- FR-10: The application must let the user add grand totals, subtotals and other type of basic aggregations.
- FR-11: The application must allow the user to modify the font style and formatting.
- FR-12: The application must let the user add different kind of graphs or charts, including the following types:

  - Bar
  - Line
  - Area
  - Pie
  - Line-bar
  - Radar
  - Waterfall

- FR-13: The application must let the user show calculated information and edit the column formula that will be shown.
- FR-14: The application will let the user to prompt to enter a set of filter criteria to be shown in the report or dashboard. This will automatically create an ad-hoc query.
- FR-15: The application shall let the user to drill down from one time hierarchy level to another: Year > Quarter > Month > Date. This will also be possible in graphs.
- FR-16: the application will let the user to navigate through different dashboards, reports and sections of OBIEE.
- FR-17: The application will be able to apply conditional formatting depending on some certain conditions.
- FR-18: The application will be capable of embedding maps with a certain level of interaction.
- FR-19: The application will allow the creation of analysis (also called reports) that will allow show any of the information or element included in the previous requirements.
- FR-20: The application will ultimately be able to create and show dashboards that will consist of a combination of reports, prompts and other elements.

4.1.2. Non-Functional Requirements

These requirements include the ones omitted in the previous section, related to quality and technical needs for the correct use of the application.

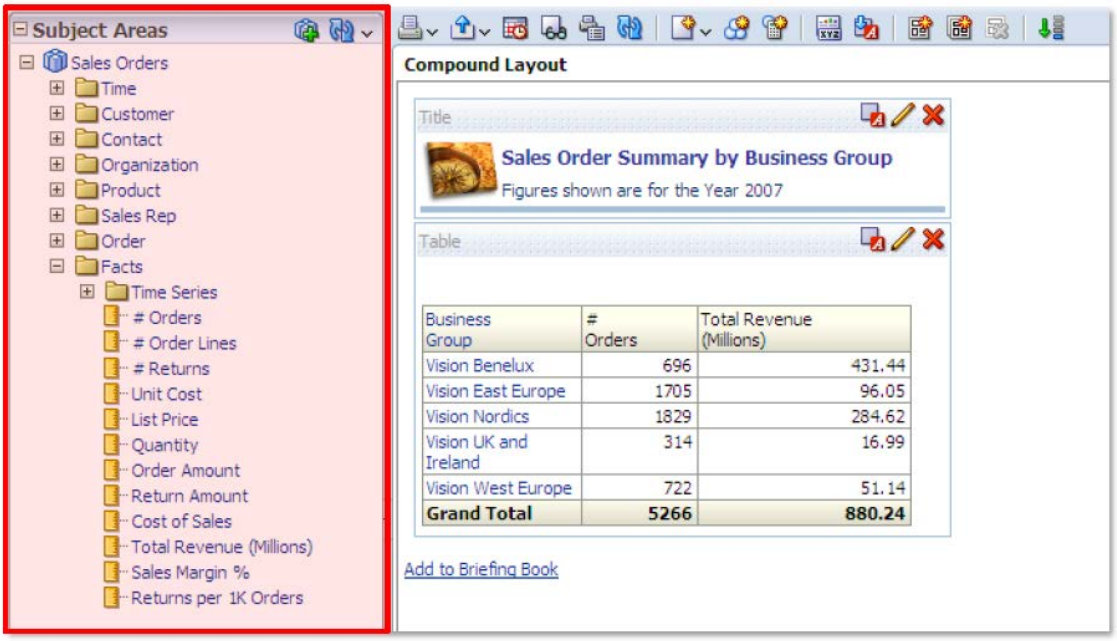
The list of Non-Functional Requirements is as follows:

- NFR-1: The operative system used by the user must be Windows XP or a later version.
- NFR-2: Minimum Available Memory required 4GB
- NFR-3: A CPU speed of at least 1GHZ
- NFR-4: Temp Space must be 270 MB
- NFR-5: Monitor must be at least of 256 colors
- NFR-6: One of the following browsers, Firefox, Chrome or IE.
- NFR-7: Must have a mouse and a keyboard.

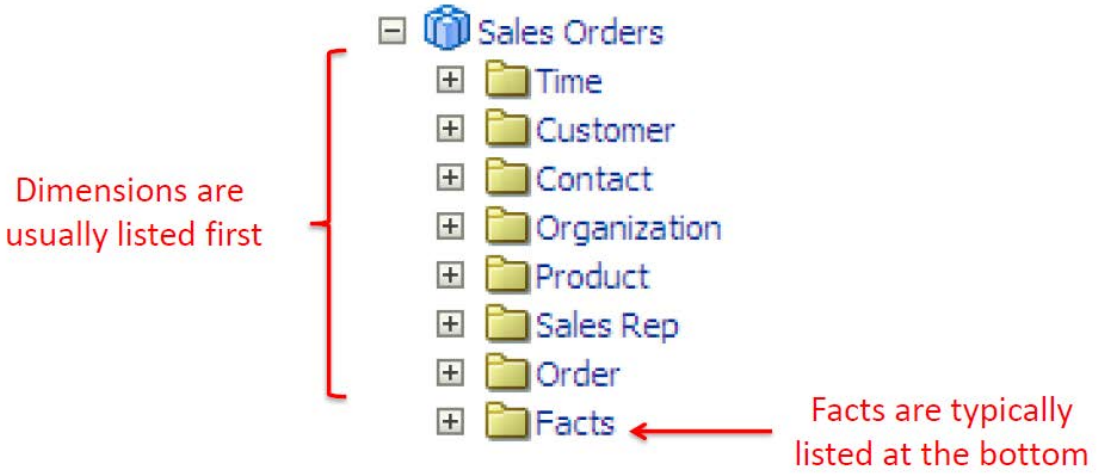
4.2. Data Modelling

Once the requirements are clear we can proceed to model the data that will be used by our application in analysis (reports) and dashboards.

In OBIEE, the term "Modelling" refers to the development of "Subject Areas" that provides Report developers with all the objects needed to build their own reports (or "Analyses"):



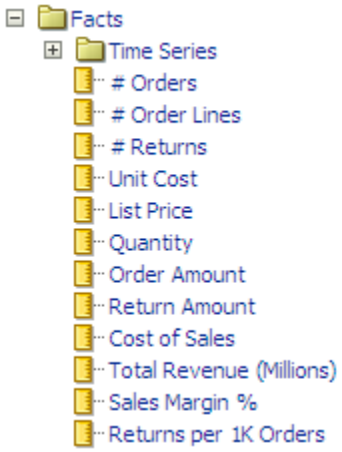
"Sales Orders" Subject Area will consist of related Facts and Dimensions that can be used to report on Sales Orders:



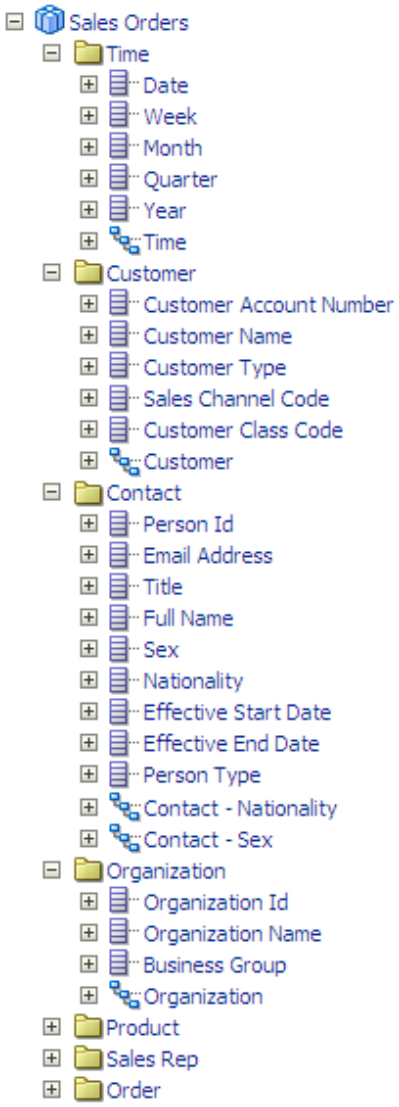
"Facts" are the numerical values that appear on reports as:

- Metrics
- Calculations
- KPIs
- Measures

Facts are defined as objects which can be aggregated (summarized). Good examples are #Orders or Total Revenue.



Dimensions are attributes that typically consist of characters or dates. Dimensions can be defined as objects which cannot be aggregated (summarized). Good examples are: Year, Date, Customer Name, Customer Type, Email Address or Organization Name.



Reports will consist of a combination of Facts and Dimensions. Often the "Fact by Dimension by Dimension" notation should be used to specify your reporting requirements. In the example below, we have 2 Facts by 1 Dimension: "# Orders" and "Total Revenue (Millions)" by "Business Group"

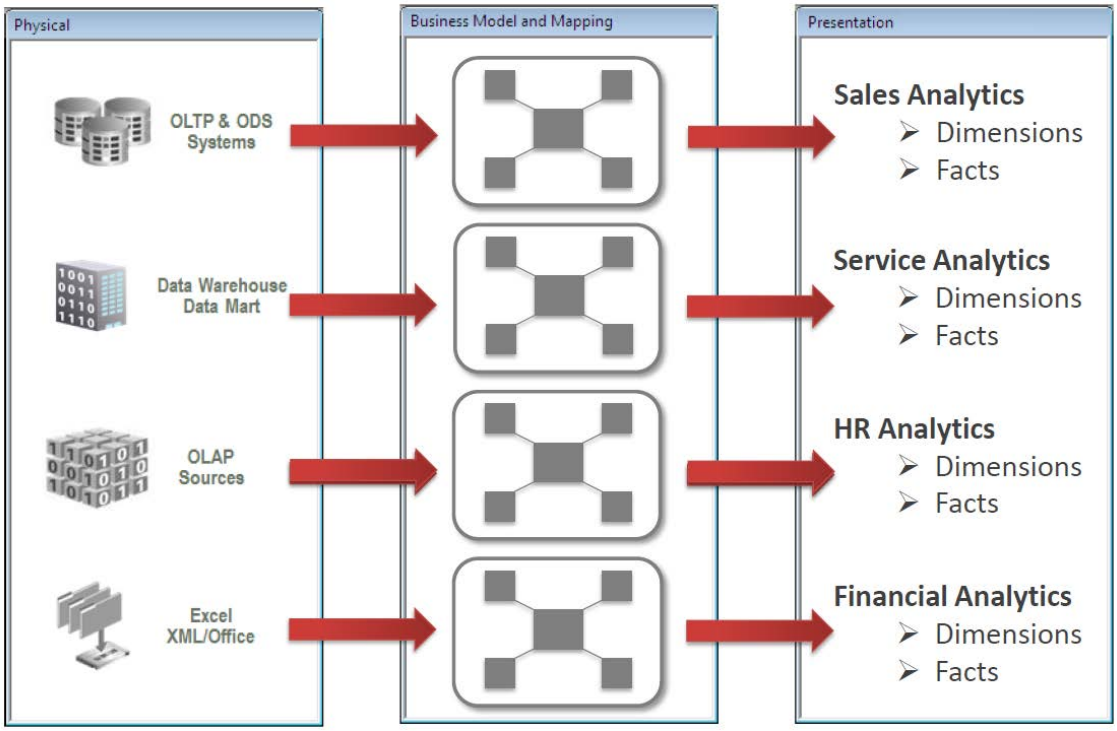
Business Group	# Orders	Total Revenue (Millions)
Vision Benelux	696	431.44
Vision East Europe	1705	96.05
Vision Nordics	1829	284.62
Vision UK and Ireland	314	16.99
Vision West Europe	722	51.14
Grand Total	**5266**	**880.24**

Note that only the Facts can be aggregated (summarised) to form a **Grand Total**:

- "# Orders" : **COUNT** aggregation rule
- "Total Revenue" : **SUM** aggregation rule

4.2.1. Modelling Principles

There are three modelling layers to consider:



In the **Physical Layer** we can model just about any type of physical data source: Relational Databases, Cube Engines (OLAP, Essbase etc), Spreadsheets, Flat Files, XML, ODBC. In addition, can handle any type of physical complexity: Third-Normal-Form (3NF), Star-Schema, Flat structures, Many-to-Many, Aggregation, Denormalisation, Outer Joins, Cross-Database Joins, Fragmented sources, Federated sources…

In the **Modelling Layer** the number, type and complexity of the underlying physical data sources does not matter. With OBIEE, it is always possible to overcome that physical complexity by translating it in to a simplified set of "Logical" Star-Schemas. Each "Logical" Star-Schema consists of one central "Logical" Fact and one or more "Logical" Dimensions.

Once you have a simplified set of "Logical" Star-Schemas, in the **Presentation Layer** you can present nice simplified sets of Dimensions and Facts to the End Users to report on.

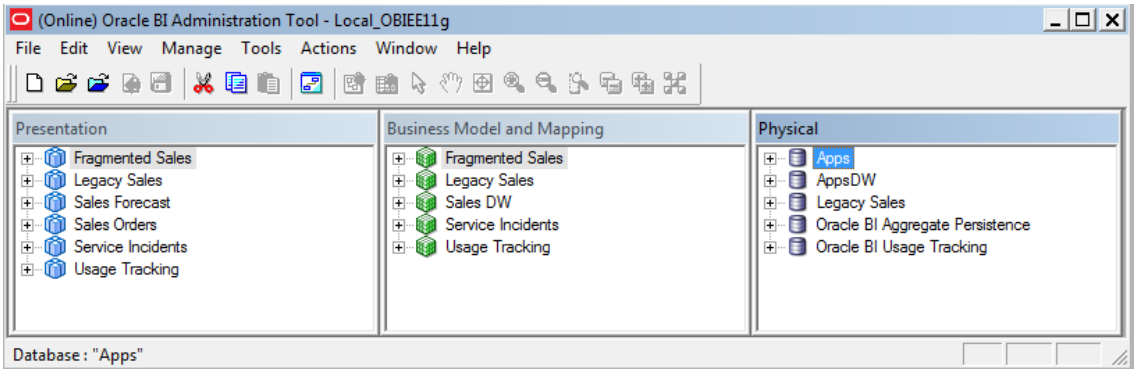
4.2.2. The BI Repository (RPD)

The BI Repository (RPD) is the file used by OBIEE to store all the metadata on the objects to be made available for reporting purposes. The RPD file is located on the OBIEE server machine. It is a single file, and will typically be between 0.25 and 10MB in size (but in some cases it can be as large as 50MB).

Development can be done in either "off-line" or "on-line" mode, however normally only a single developer should ever be modifying the RPD in "on-line" mode at any one time. For environments where multiple RPD developers are working simultaneously, "Multi-User-Development" (MUD) should be used.

The RPD consists of 3 layers:

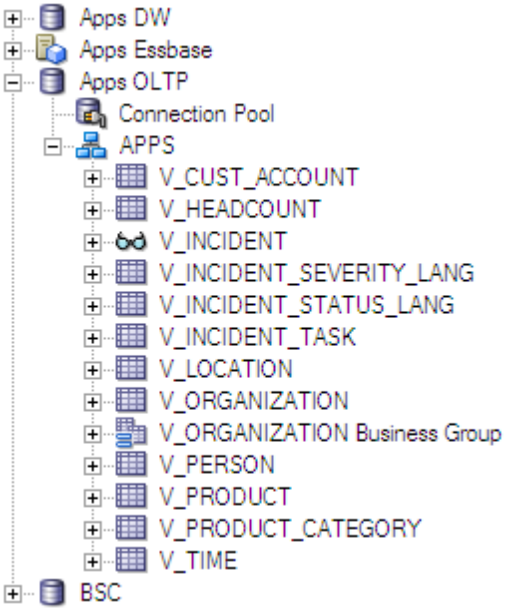
- Physical Layer: Defines the physical database structures and their relationships
- Business Model Layer: One or more "logical" models that translate the more complex physical data structures in to a simplified set of business-oriented structures ("Logical" Star-Schemas)
- Presentation Layer: Related sets of reporting objects are presented to the user via "Subject Areas"



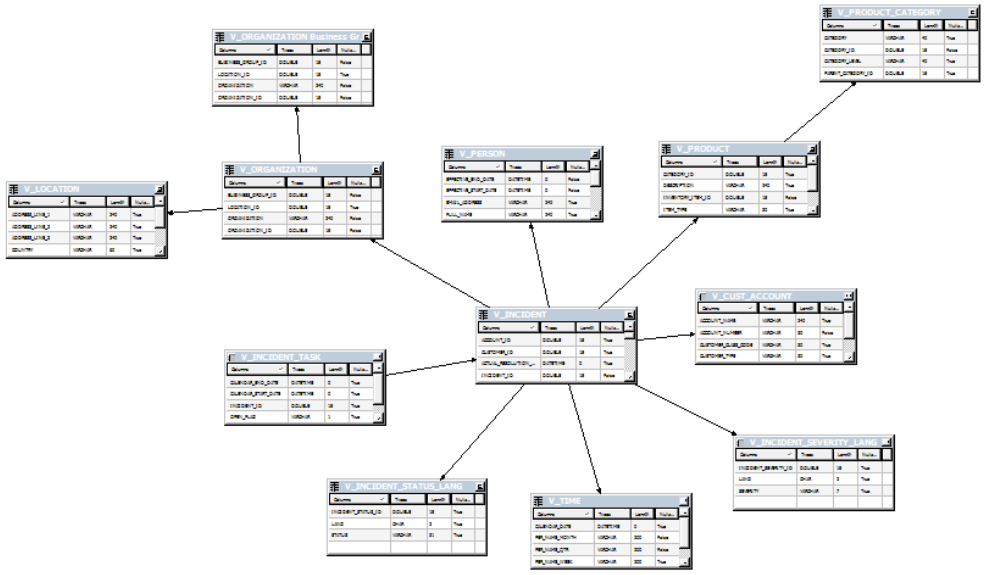
4.2.2.1. The Physical Layer

The first modelling task is to import the Physical objects. The Physical Layer may consist of one or more "Relational" databases. There may also be "Cubes" or "Multi-Dimensional" data sources (e.g. Essbase).

Within each Physical Database there is a "Connection Pool" to define the connection settings to the corresponding data source: Data Source Type, Username, Password, Maximum Connections, Connection Pooling (Sharing)…

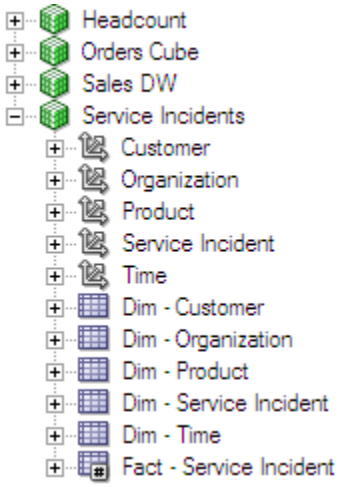


For a "Relational" Physical Database, we will also define how all the tables join to each other. We can import the joins automatically when you import the tables in to the RPD. We can also create new joins (compound and complex joins are possible). The joins are One-to-Many (1:M) relationships.



4.2.2.2. The Modelling Layer

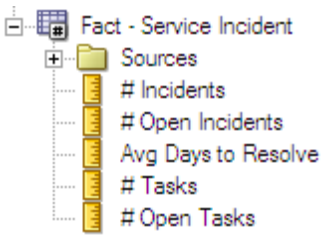
The Business Model Layer will consist of one or more "Business Models". A Business Model can consist of objects sourced from multiple Physical Databases. It is possible to combine all the reporting objects in to a single Business Model. However, we are going to create separate Business Models since we have sets of reporting objects that serve completely separate business functions with little or no overlap. For example: Real-time "operational" reporting and Data Warehouse "historic" reporting.



Each and every Business Model will consist of the following types of object:

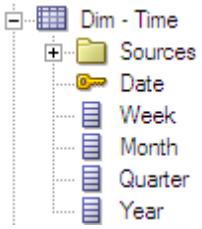
- Logical Dimension Table: A set of related Logical Dimension Columns e.g. "Customer Name"
- Logical Fact Table: A set of related Logical Fact Columns e.g. "Total Revenue"
- Dimension Hierarchy: Defines a Dimension Hierarchy structure e.g. Year > Quarter > Month > Date

We will start by modelling your Logical Fact Table first. This will contain the relevant metrics for the new Logical Star Schema. A Logical Fact Table should only consist of **Logical Fact Columns**, numerical values that have an "aggregation rule" assigned. For example: SUM, COUNT, MIN, MAX.

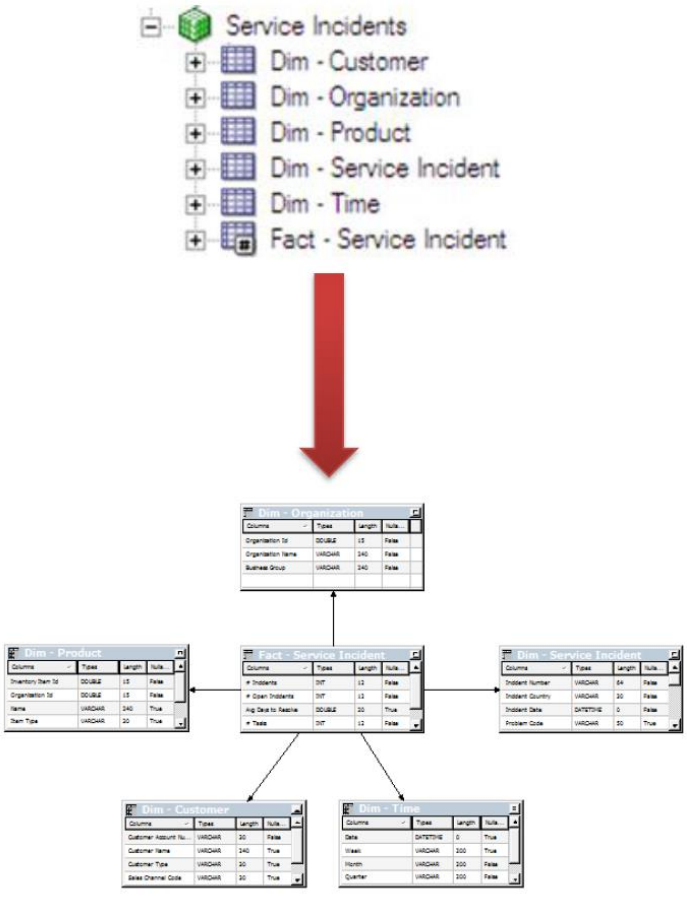


We can then start modelling the related Logical Dimension Tables. A Logical Dimension Table should only consist of:

- Logical Keys: Each Dimension Logical Table must have at least one Logical Key to state which Logical Column(s) provide uniqueness.
- Logical Dimension Columns: Typically character or date values. These never have an "aggregation rule" assigned as they cannot be aggregated (summarized).



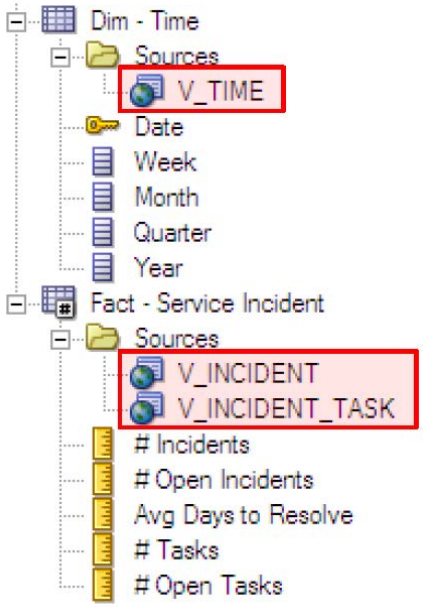
When all your Logical Tables are modelled, we create "Logical Joins" to define the relationships between them. The relationships will form "Logical Star-Schemas" where a central "Fact" is surrounded by one or more "Dimensions". Logical Joins simply state that there is a 1:M relationship between two Logical Tables. We don't actually define how the Logical Tables join together, the BI Server decides this for itself at run-time. As a strict rule to avoid "Snow-Flaking": Facts join only to Dimensions and Dimensions join only to Facts.



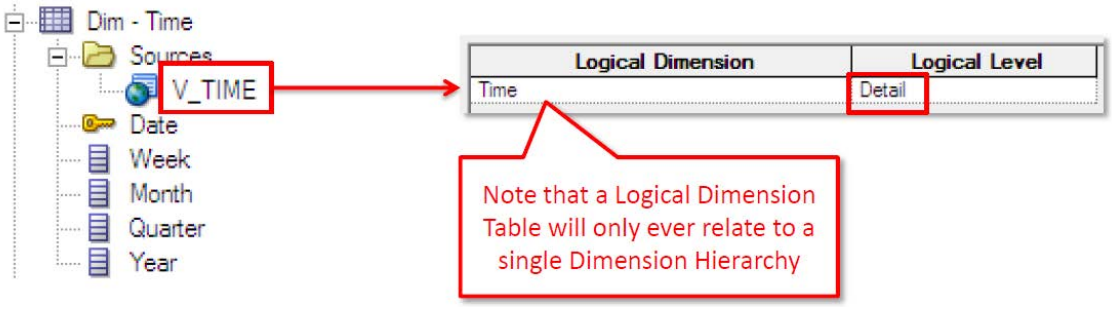
For every Logical Dimension Table, a Dimension Hierarchy is created. Dimension Hierarchies are extremely useful for providing users with automatic "drill downs" when viewing reports (Analyses) within the BI Dashboards.

Logical Table Sources (LTS) are one of the most significant features of the Business Model, with a variety of uses:

- They contain the definitions of each Logical Column, specifying any transformations that may occur between Physical/Logical models.
- LTS can consist of multiple physical tables – grouping multiple tables together in to a single LTS is an important feature when simplifying complex physical data-models.
- They enable sophisticated features such as "Aggregate Navigation" and "Fragmentation".



We need to set the "Content Levels" for every LTS. The concept of Content Levels can take time to understand, but they have important users. For each LTS, they state which Dimension Hierarchies they have a relationship with and at what levels. OBIEE uses the metadata in its decision making so that it knows the best way to optimize and parse queries.

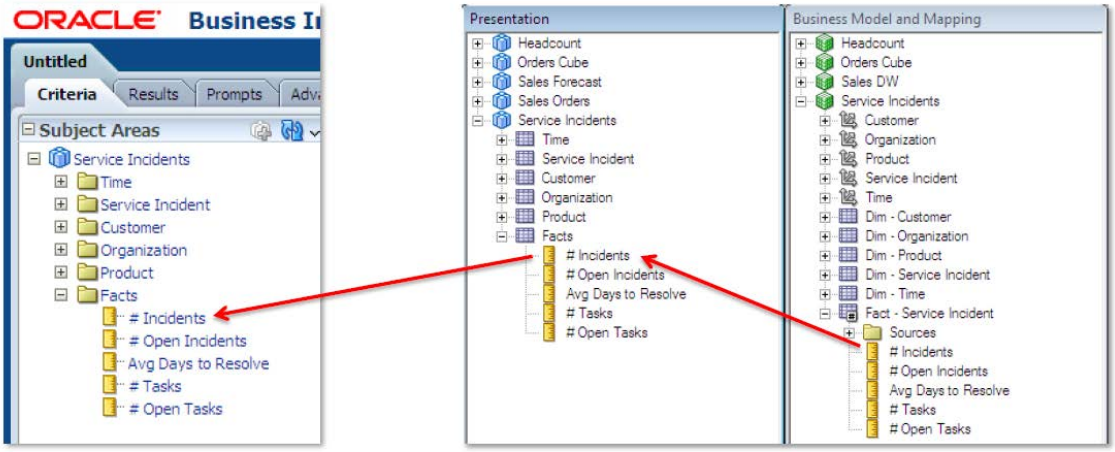


4.2.2.3. The Presentation Layer

The Presentation Layer is the simplest of the three layers within the BI Repository (RPD), but it is still very important since this is the layer with which the End Users interact.

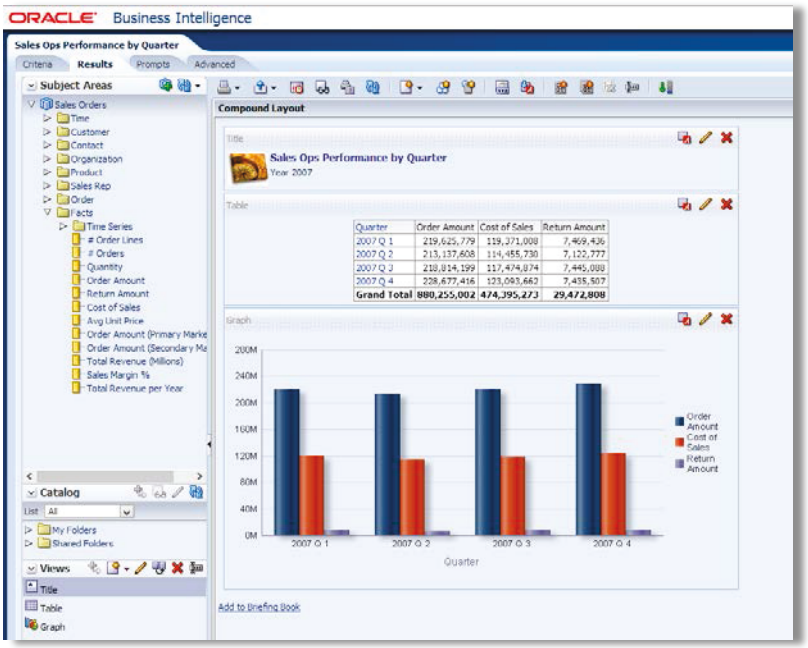
The Presentation Layer is used to define the "Subject Areas" that are available within Answers:

A Subject Area will consist of Presentation Tables and Presentation Columns. Presentation Columns will map to a single Fact or Dimension Logical Column within the Business Model Layer.

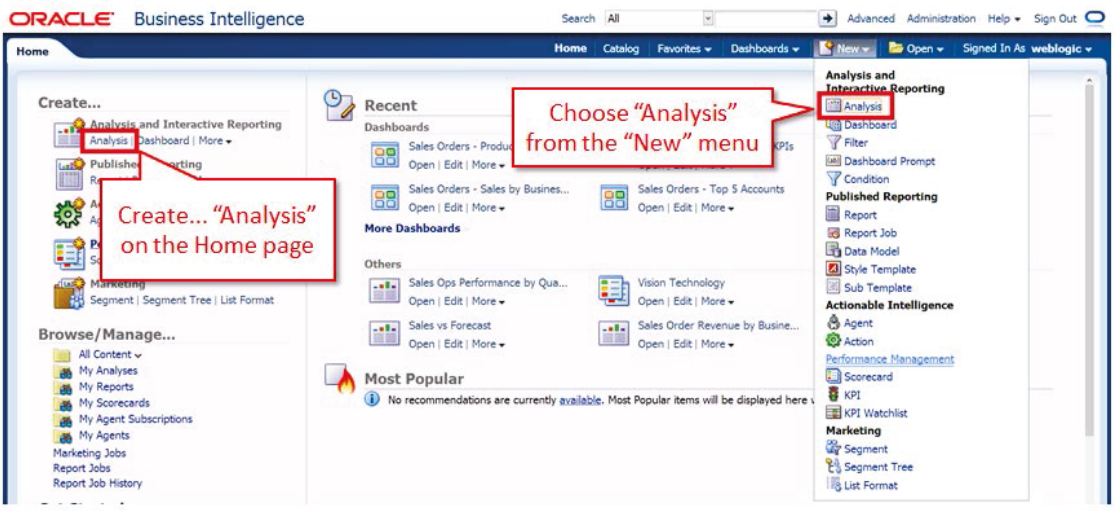


4.3. Design of an analysis or report

At this point we already have a Subject area created and uploaded to OBIEE. Now we will use Answers to create reports. Answers is the component of Oracle BI used for building interactive reports (referred to as “Analyses”).

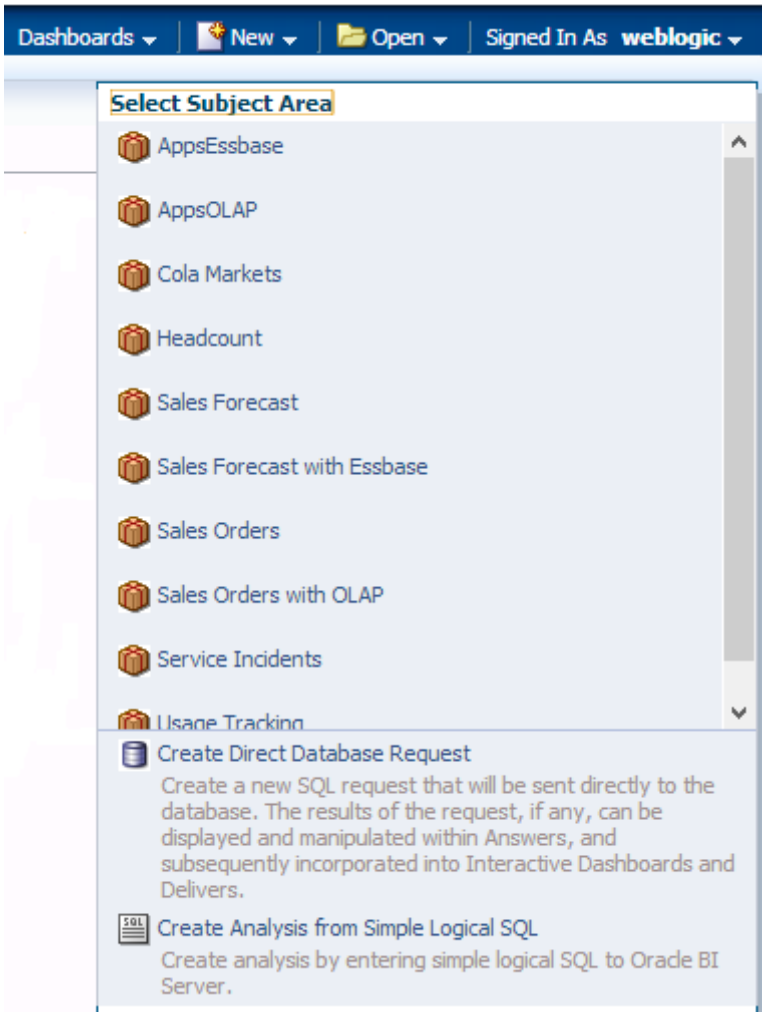


We create a new “Analysis” in two ways.

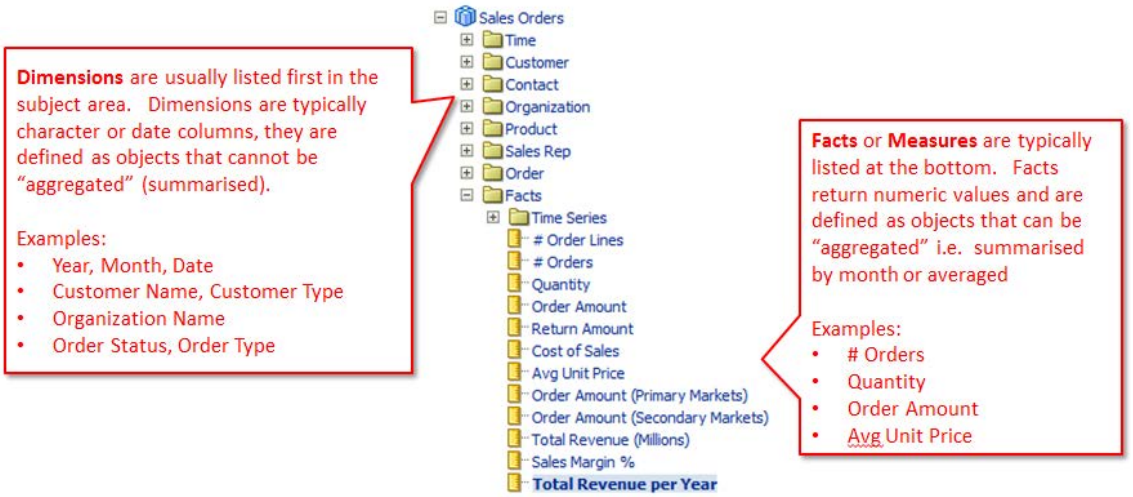


When we choose to create a new Analysis, we will be asked to select a "Subject Area".

Subject Areas contain related sets of reporting objects, like the one we created in the previous section. For example, the "Sales Orders" Subject Area contains objects relevant to reporting on sales orders: Year, Organization, Customer, Total Revenue, Cost of Sales…

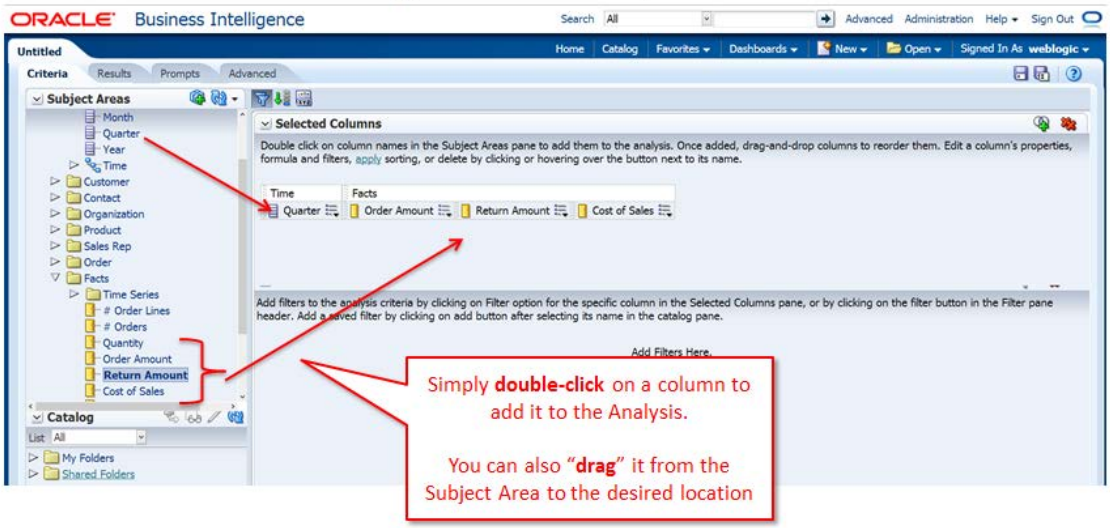


A Subject Area consists of "Facts" and "Dimensions". To create an Analysis, we will select a combination of both Facts and Dimensions to appear on it.

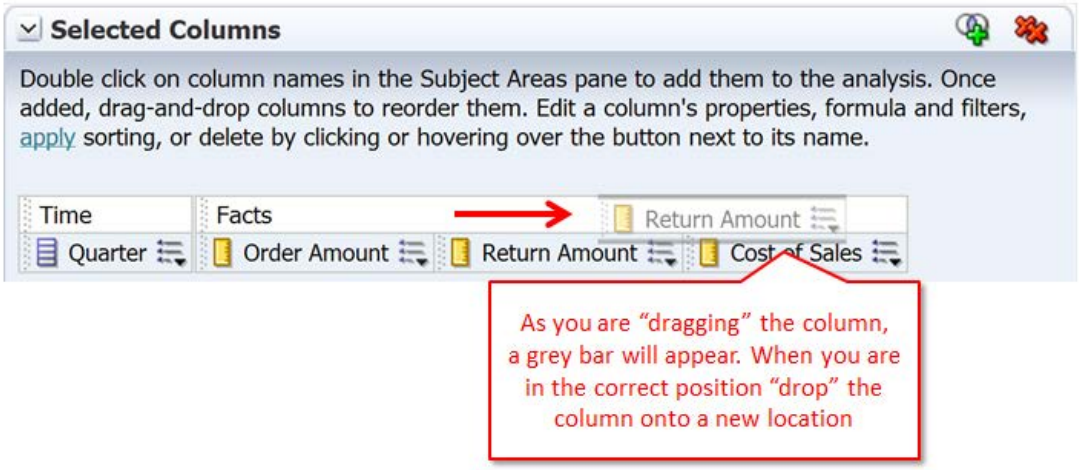


Once we have chosen the Subject Area, we will be taken to the Answers "Criteria" tab where we define the data content of your Analysis.

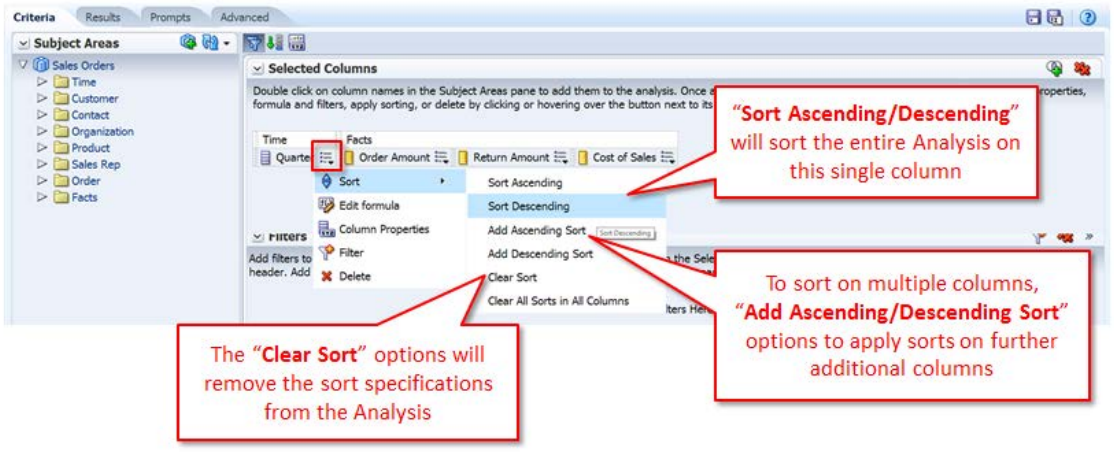
The first step is to choose the desired columns from the Subject Area which should appear on the Analysis.



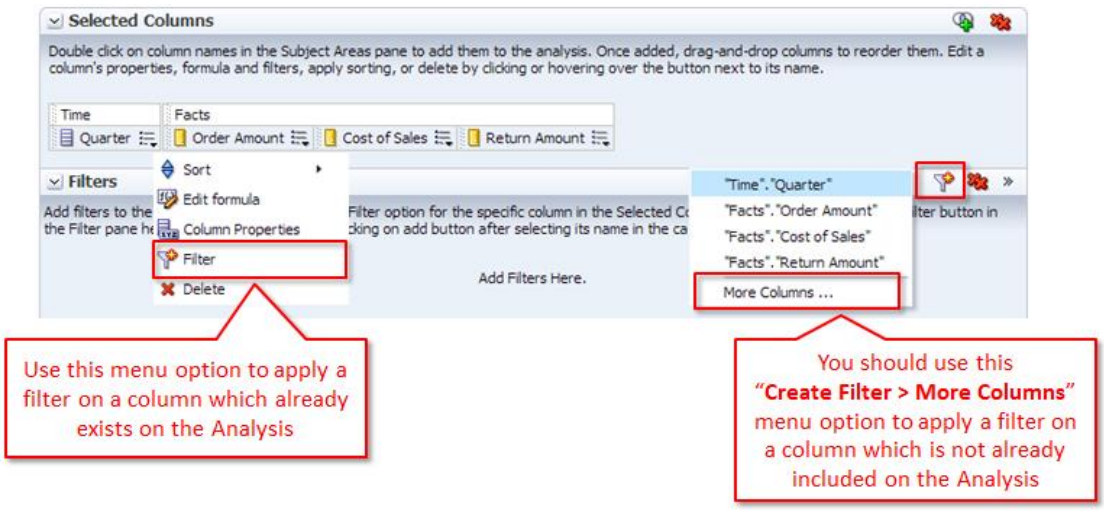
If the columns are not listed in the correct order, we can rearrange them simply by dragging and dropping them onto the desired location:



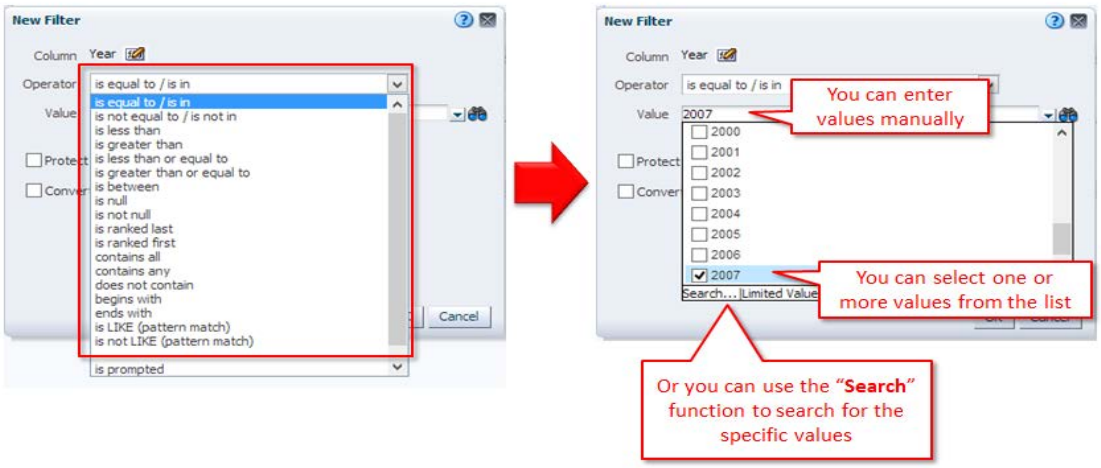
By default, the Analysis is sorted alphabetically starting with the first column, however we can choose a different sort order using the menu beside each column selected.



Filters can be applied to limit the results returned by your Analysis.

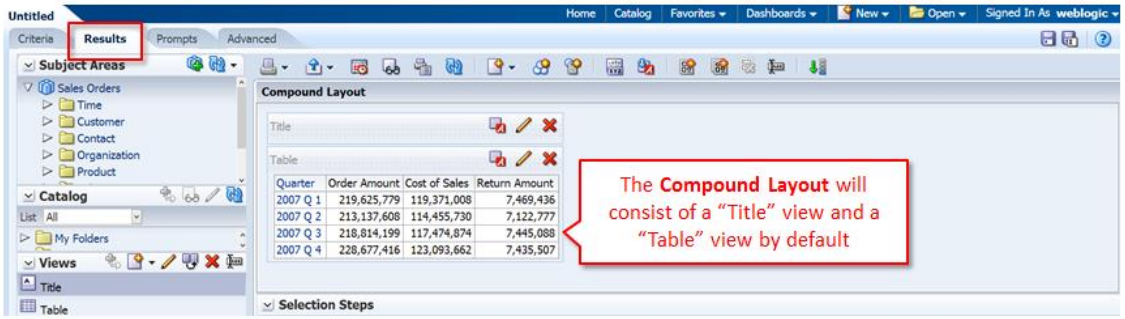


Once we have selected the column on which to filter, specifying the filter criteria is easy:

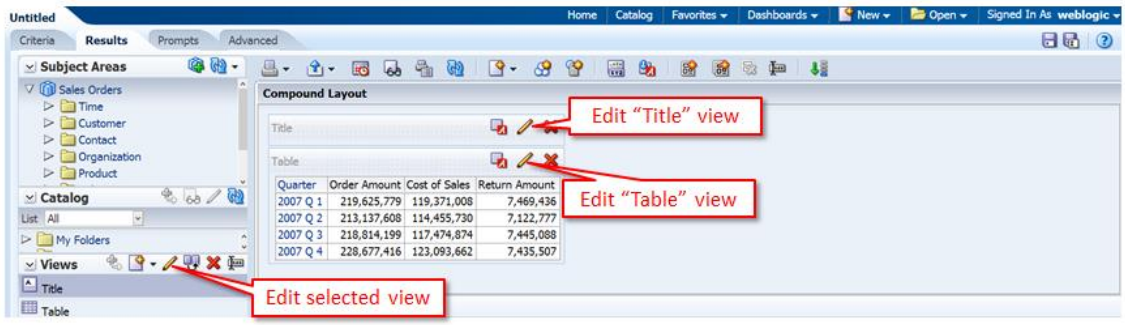


Once the content of the request has been defined, we go to the "**Results**" tab to see the actual results returned.

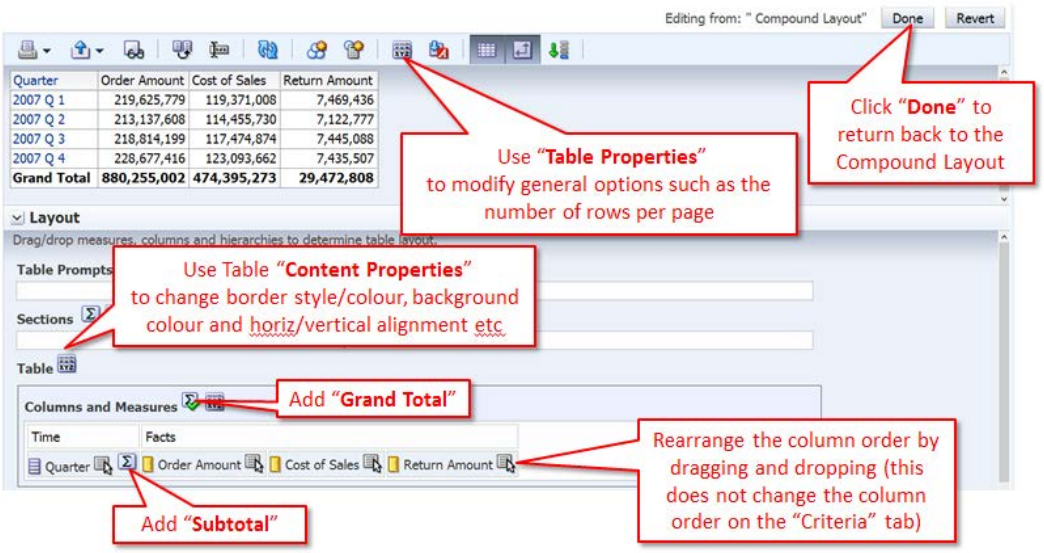
We will be shown the "**Compound Layout**" which is the default view of the request when it is opened on a dashboard.



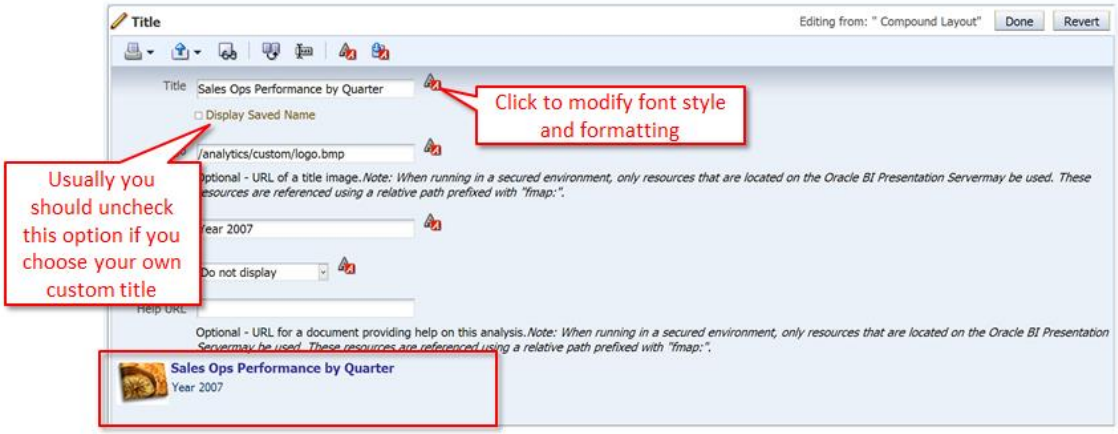
We edit views simply by clicking on the "edit" buttons either in the "Views" pane or at the top of each view in the Compound Layout:



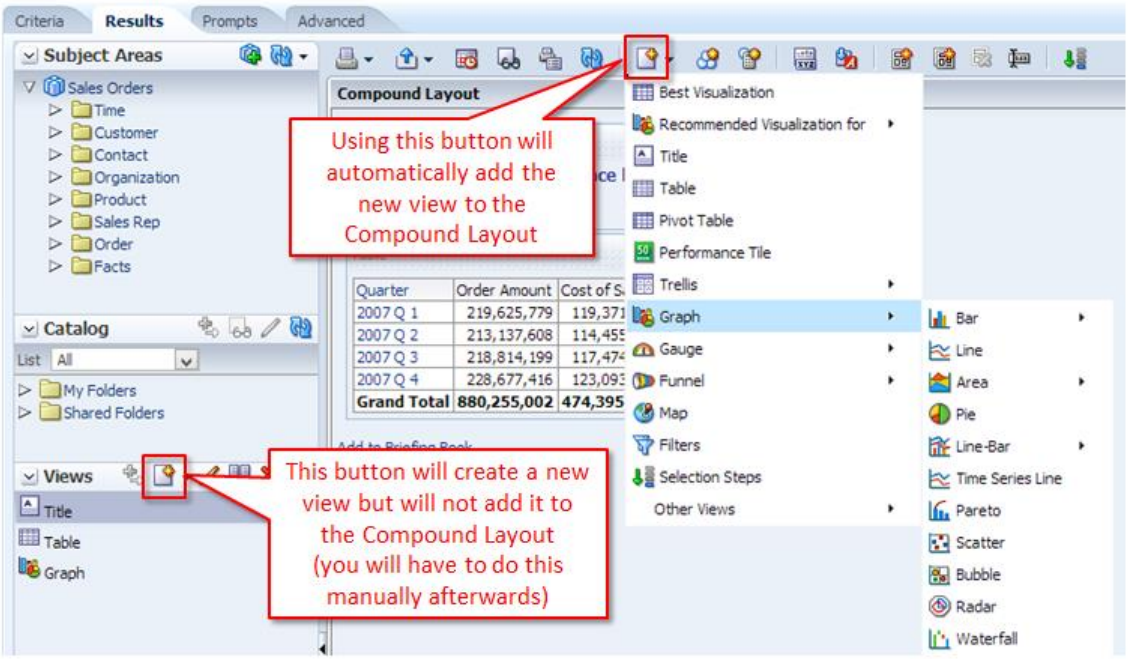
Editing a Table view allows you to add Grand Totals and modify the layout and formatting.



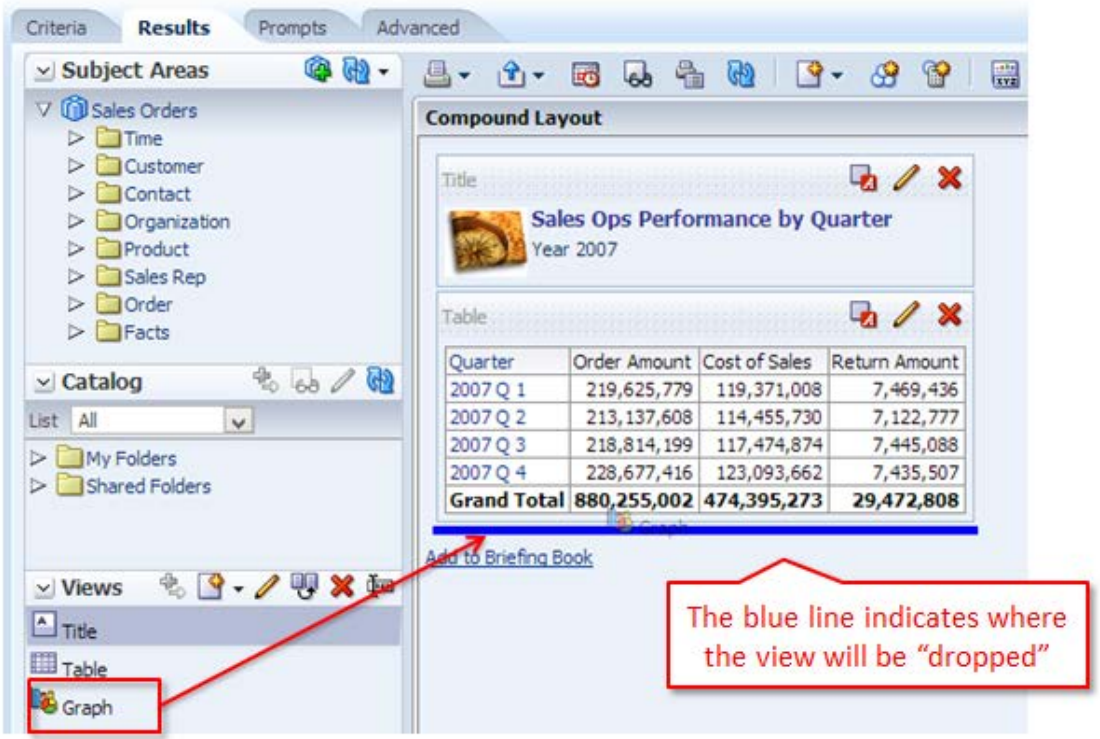
We then edit the Title view to configure items such as the Title and Subtitle.



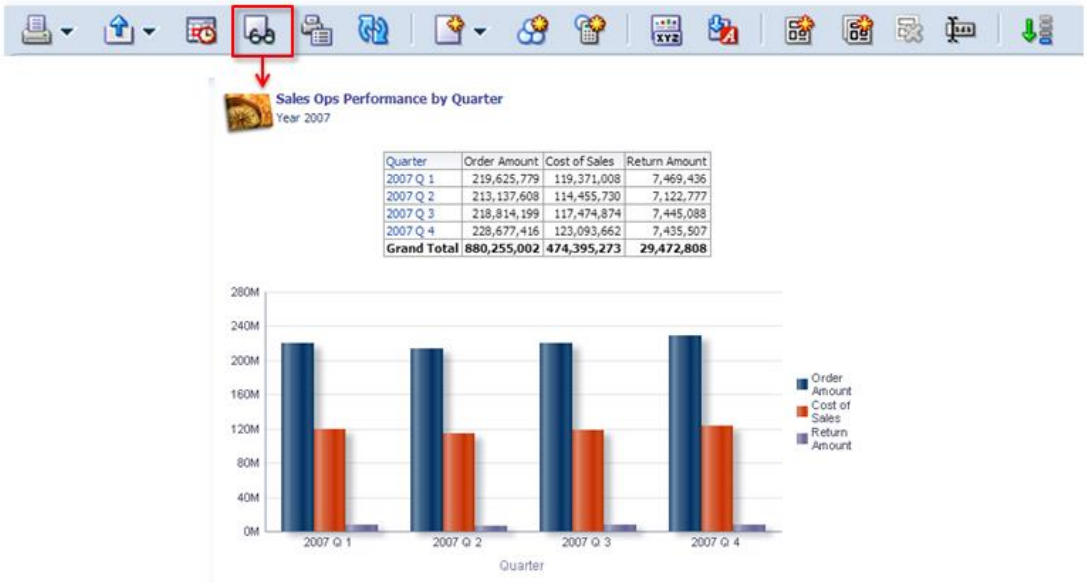
We add a new view to the Analysis using the "new" button which exists in two places.



We drag views from the "Views" window pane directly onto the Compound Layout:

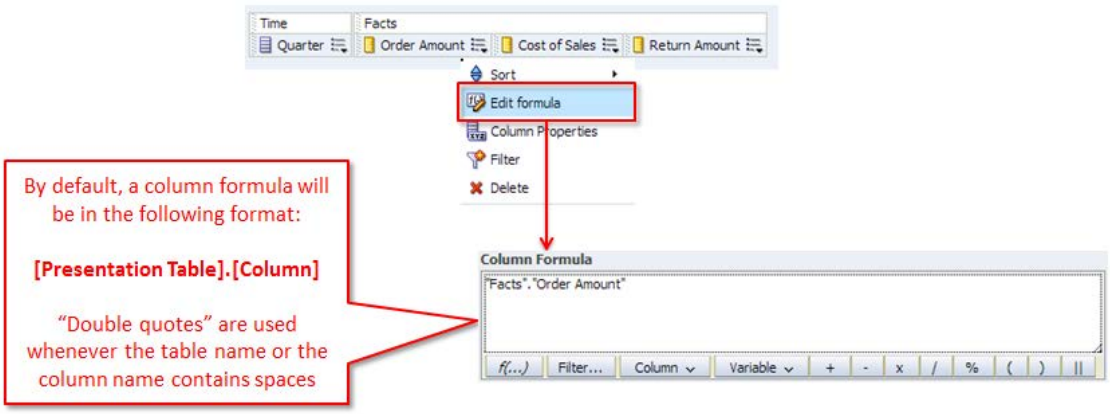


And finally, the "**Preview**" option shows the Analysis exactly as it would appear on a Dashboard:

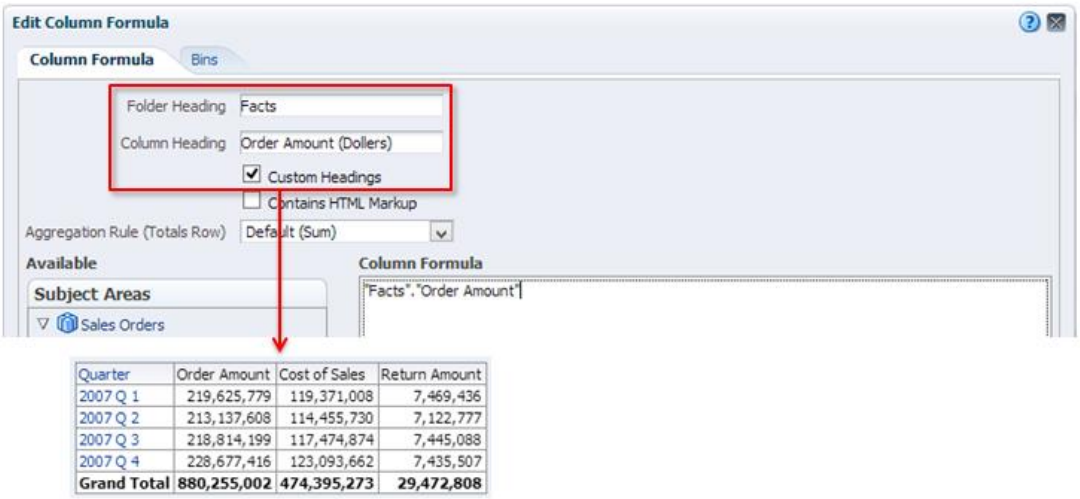


4.4. Column Formulas

As per the functional requirements specification it is needed to perform additional calculations or formulas to the objects present on the request. In order to achieve this, we edit a column's formula by choosing the "Edit Formula" menu option on the "Criteria" tab:



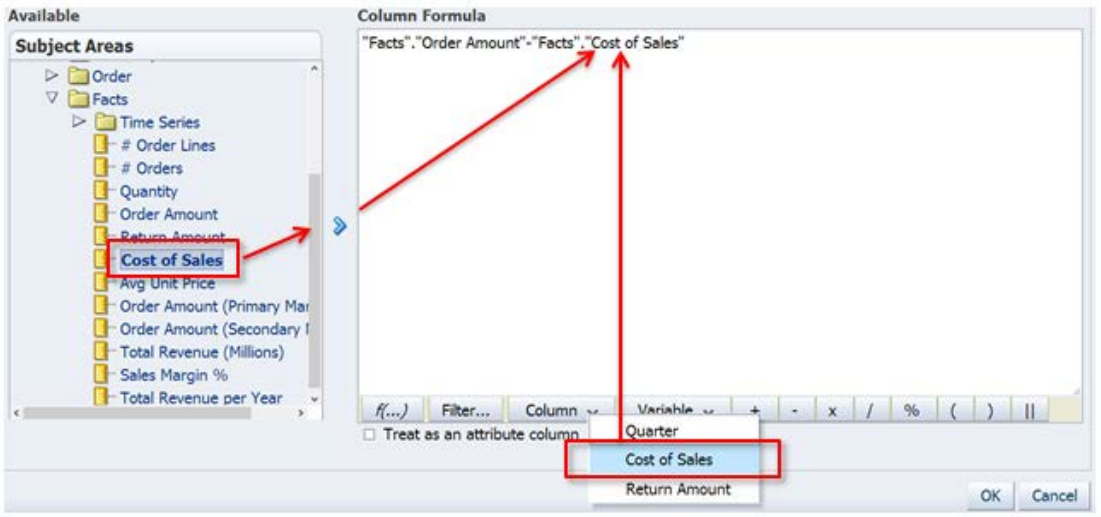
The "Edit Formula" options allow us to override the default Table / Column Heading



Quarter	Order Amount	Cost of Sales	Return Amount
2007 Q 1	219,625,779	119,371,008	7,469,436
2007 Q 2	213,137,608	114,455,730	7,122,777
2007 Q 3	218,814,199	117,474,874	7,445,088
2007 Q 4	228,677,416	123,093,662	7,435,507
Grand Total	**880,255,002**	**474,395,273**	**29,472,808**

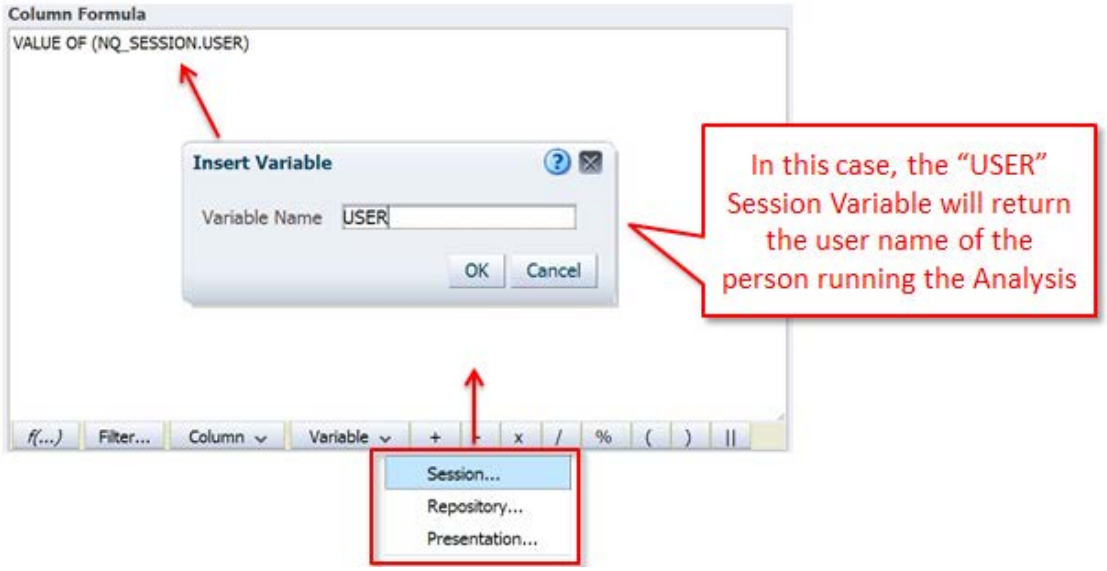
We enter new expressions and formulas manually by typing.

There are also shortcuts for adding new columns into the expression:



The "Variables" button is used to add Session, Repository or Presentation variables to your Analysis.

Variables can simplify your expressions and make them more dynamic (less hard-coding) by making use of pre-calculated variables such as "CURRENT_YEAR" .



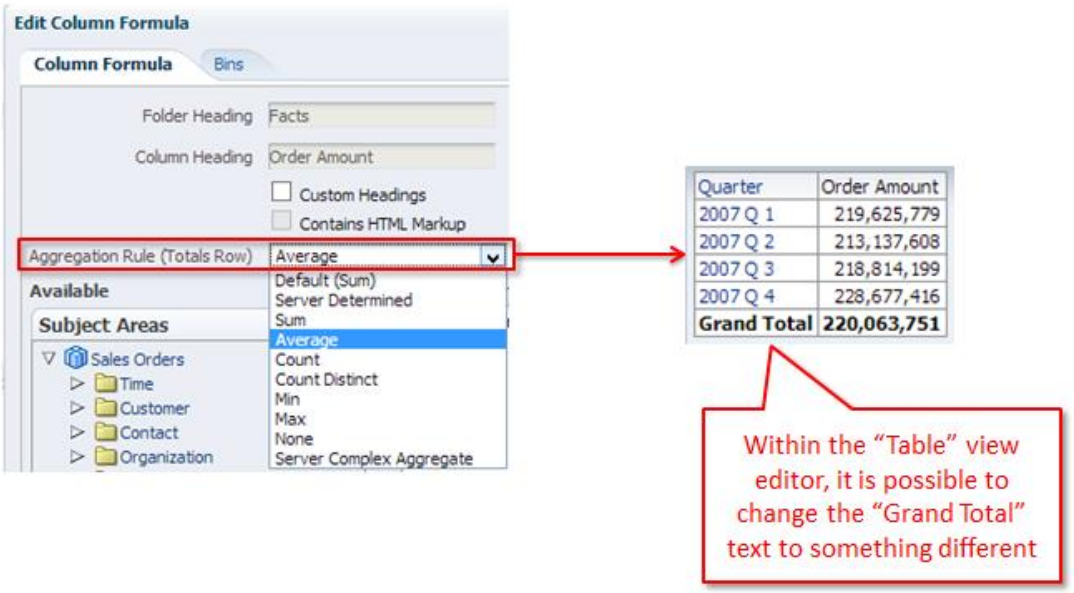
There are 3 types of variable which can be created:

- **Session Variables** are variables that are unique to each user's session. When a user logs in, a process takes place to "initialise" all their Session Variables. For example:
  - USER: Your OBIEE user name e.g. ASMITH
  - MY_ORG: The name of your own specific Organization
- **Repository Variables** are global variables where all users see the same value (they are not unique to each user's session). For example:

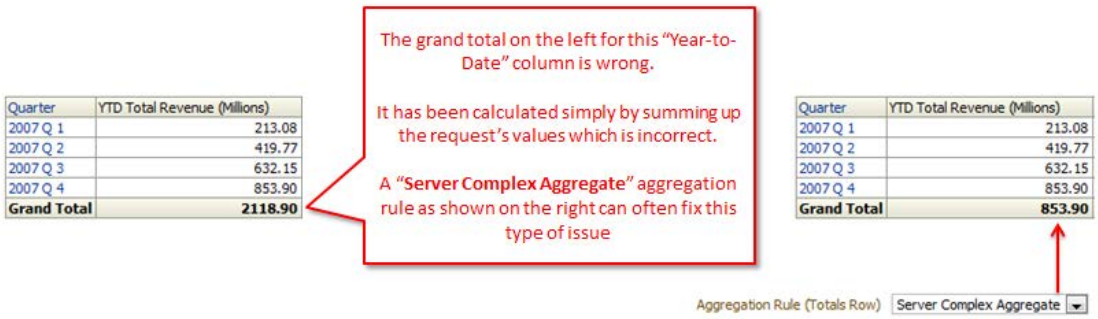
- YEAR_AGO_CAL: The calendar date 1 year ago
- CURRENT_YEAR: The current year e.g. 2011

- **Presentation Variables** are variables that can be created by report developers and be referenced in their reports. Users are able to change the value assigned to a Presentation Variable using a "Dashboard Prompt". For example, a user could select the Year "2010" in a Dashboard Prompt and the text "Year 2010" would appear in the report's subtitle.

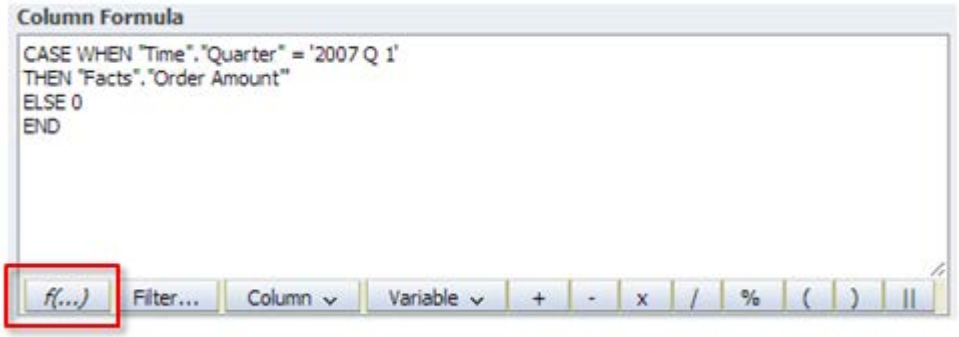
We can override the default "Aggregation Rule" which determines how the grand total and sub totals are calculated

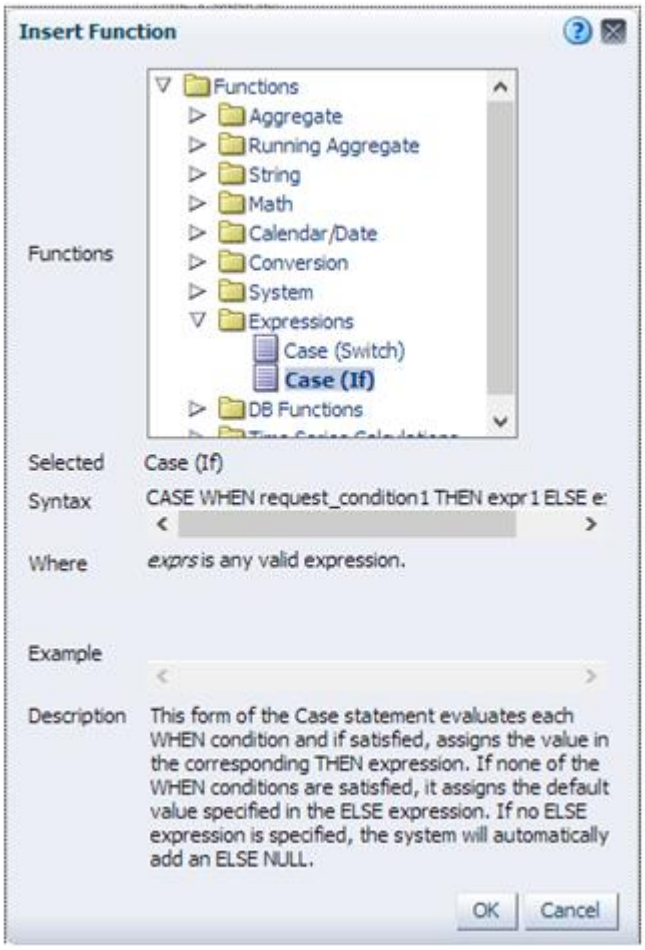


Sometimes it is not possible or logical to calculate the grand total by summarising the values displayed within the Analysis. If we find that grand total has not been calculated correctly, then we set the aggregation rule to "**Server Complex Aggregate**". The "Server Complex Aggregate" setting will force OBIEE to issue a separate query to calculate the grand total, instead of trying to calculate it by summarising the end results returned the Analysis.



Use the "**Function**" button utility to build a variety of functions and expressions.



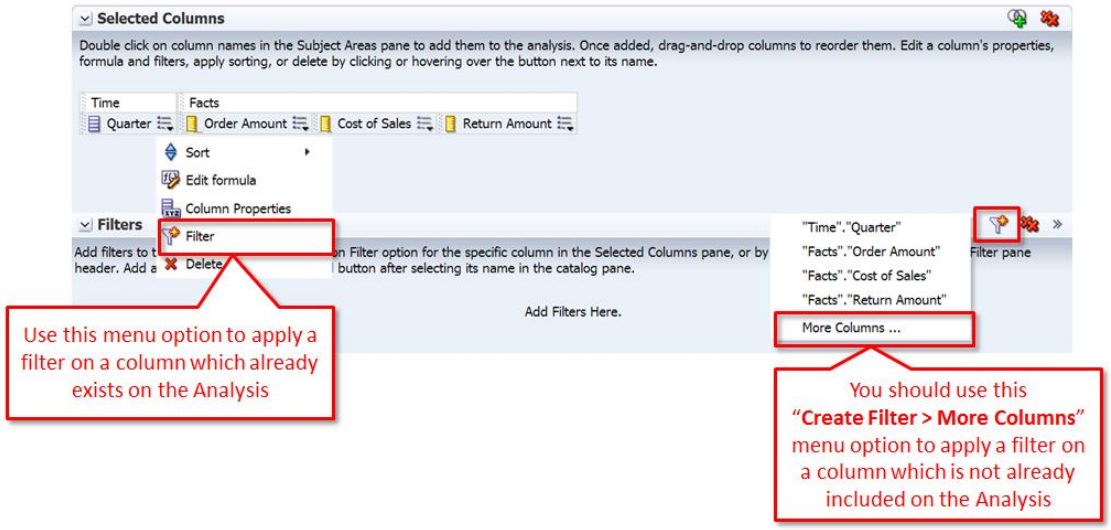


Here we have some examples:

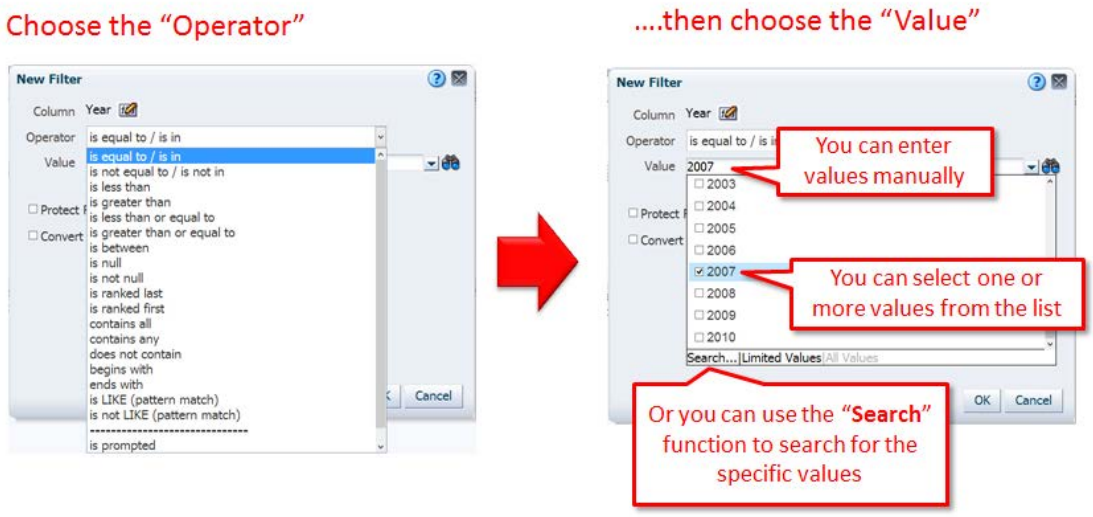
Year	Quarter	Total Revenue (Millions)	YTD Total Revenue
2007	2007 Q 1	212.16	24.9%
	2007 Q 2	206.01	24.2%
	2007 Q 3	211.37	24.8%
	2007 Q 4	221.24	26.0%
2007 Total		**850.78**	**100.0%**

4.5. Filters

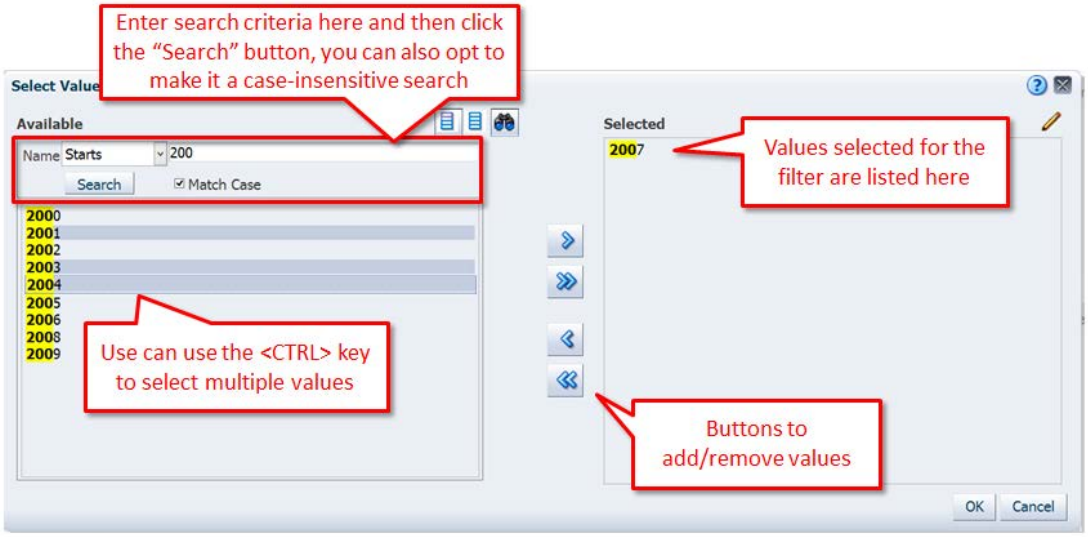
Filters are applied to limit the results returned by the Analysis and are requested in the functional requirements specification.



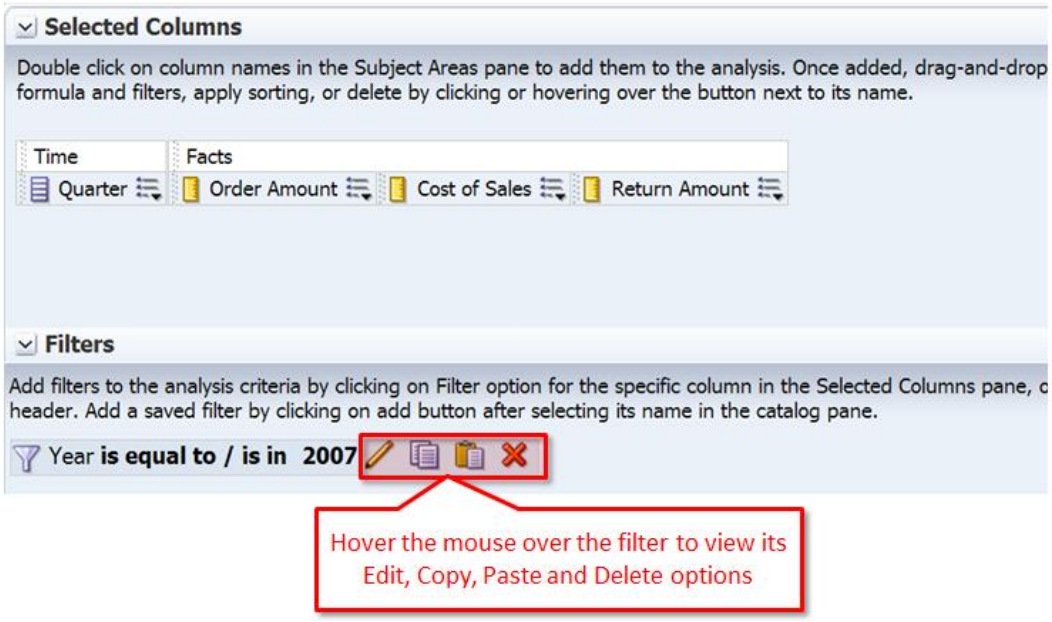
Once we have selected the column on which to filter, we specify the filter criteria:



The "**Search**" facility is straight forward to use for selecting the filter criteria:



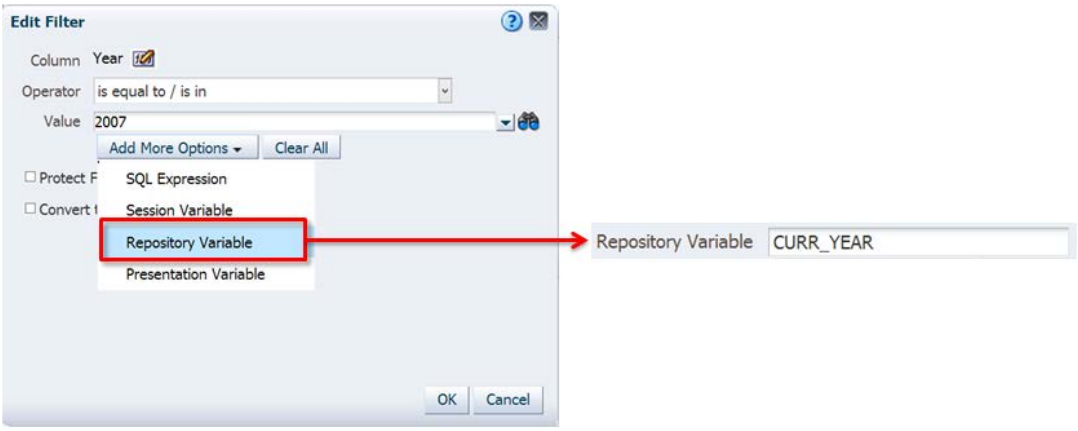
Once defined, we see all the filters listed in the "Filters" pane at the bottom of the Criteria tab.



4.5.1. Variable filters

It is desirable in many situations to filter based on a "variable" rather than a static value. For example, if we want a request to show results for the current year then we do not want to filter on "Year = 2010" as it would require to update the request every year. Instead, we choose to filter the request based on a variable. For example:

Year = VALUEOF(CURR_YEAR)



There are 3 types of variable which can be created by Developers:

(Note these variables were explained before in section 4.4. Column formulas, but due to their relevance, it is repeated here)

- **Session Variables** are variables that are unique to each user's session. When a user logs in, a process takes place to "initialise" all their Session Variables. For example:

USER: Your OBIEE user name e.g. ASMITH

MY_ORG: The name of your own specific Organization

- **Repository Variables** are global variables where all users see the same value (they are not unique to each user's session). For example:

YEAR_AGO_CAL: The calendar date 1 year ago

CURR_YEAR: The current year e.g. 2011

- **Presentation Variables** are variables that can be created by report developers and referenced in their reports. Users are able to change the value assigned to a Presentation Variable using a "Dashboard Prompt". For example, a user could select the Year "2010" in a Dashboard Prompt and the text "Year 2010" would appear in the report's subtitle.

4.5.2. SQL in filters

Very occasionally the filter criteria we need to specify is too advanced for the standard wizard, so it is possible to manually provide the SQL expression. You can do this by enabling the "**Convert this filter to SQL**" option and then clicking the "OK" button.

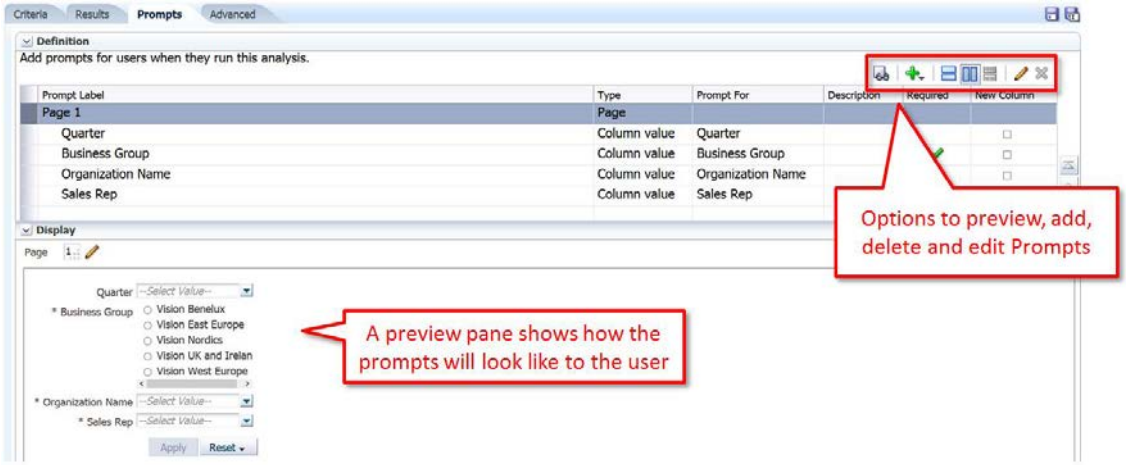
4.5.3. Prompts

As per the functional requirements, the user needs to be able to enter a set of filter criteria before they view an individual Analysis. Consider the example below where the user has opened a "detail" Analysis within Answers, 1000s of records are returned since hardly any filter criteria has been applied. This could result in serious performance issues across the system.

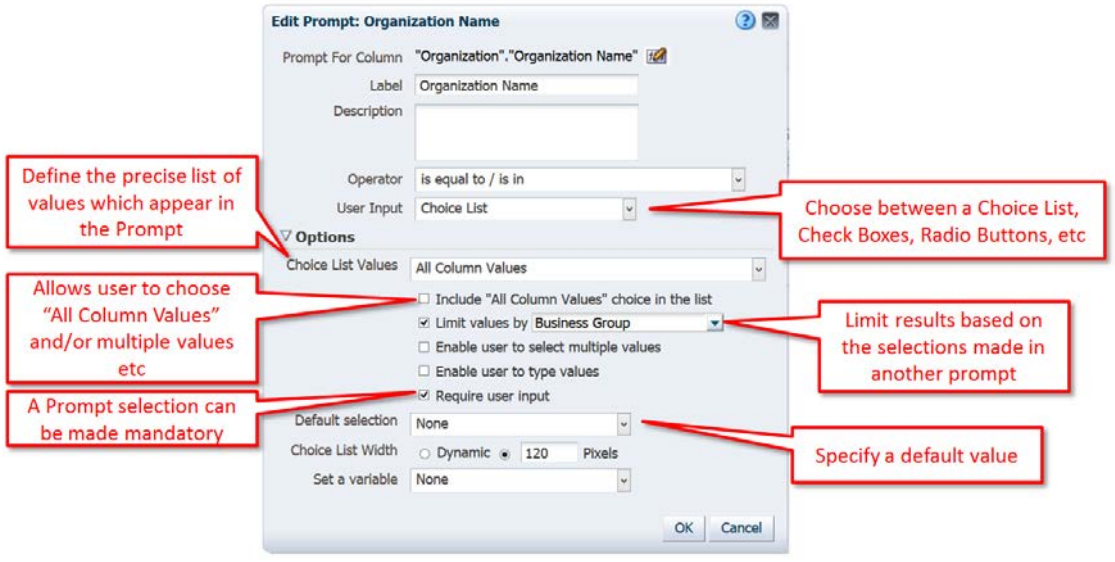
Sales Order Details by Sales Rep
Time run: 4/24/2013 7:51:41 PM
Vision Ireland; Vision UK and Ireland

Sales Rep	Customer Name	Account Number	Order Number	Deal Size	Order Date	Order Type	Order Status	Product Category	Order Amount	Return Amount
Derek, Ms. Amy	%	1928	200721	50K - 75K	22-02-2007	ORDER	CLOSED	Capacitors	29,874	2,297
								Discretes	3,150	
								Electronic Components	33,024	2,297
								Gates	3,150	
								Passives	29,874	2,297
								Product Requirements	31,468	4,198
								Service Plans	1,575	
								Vision High-Tech	66,067	6,496
	% Total								**198,181**	**17,586**
	2180	2038	250700	0K - 10K	16-01-2007	ORDER	CLOSED	Cable Assemblies	337	21
								Capacitors	3,953	558
								Electronic Components	8,226	575
								Interconnect Components	337	21
								LED's	4,273	16
								Optoelectronics	4,273	16
								Passives	3,953	558
								Vision High-Tech	8,564	596
	2180 Total								**33,917**	**2,363**
	ABC Corporation Worldwide	2575	200733	100K+	10-03-2007	ORDER	CLOSED	Connectors	49,137	
								Electromechanical Components	34,161	
								Interconnect Components	49,137	
								Power Components	34,161	
								Power Supplies	34,161	
								Product Requirements	25,762	
								Vision High-Tech	109,061	
	ABC Corporation Worldwide Total								**335,581**	
	Acme Distribution Incorporated	5731	250705	★★☆☆☆ 25K - 50K	07-02-2007	ORDER	CLOSED	Capacitors	16,235	
								Computer Parts and Components	10,500	

We then define Analysis Prompts on the "Prompts" tab within Answers. Analysis Prompts request the user to enter filter criteria before the Analysis is actually run. In the example below, the user will be prompted to filter on 4 columns:



There are a number of possible settings for each Prompt:



4.5.4. The result

When the user now opens the Analysis, a series of Prompts will make sure the Analysis now only returns an appropriate number of records:

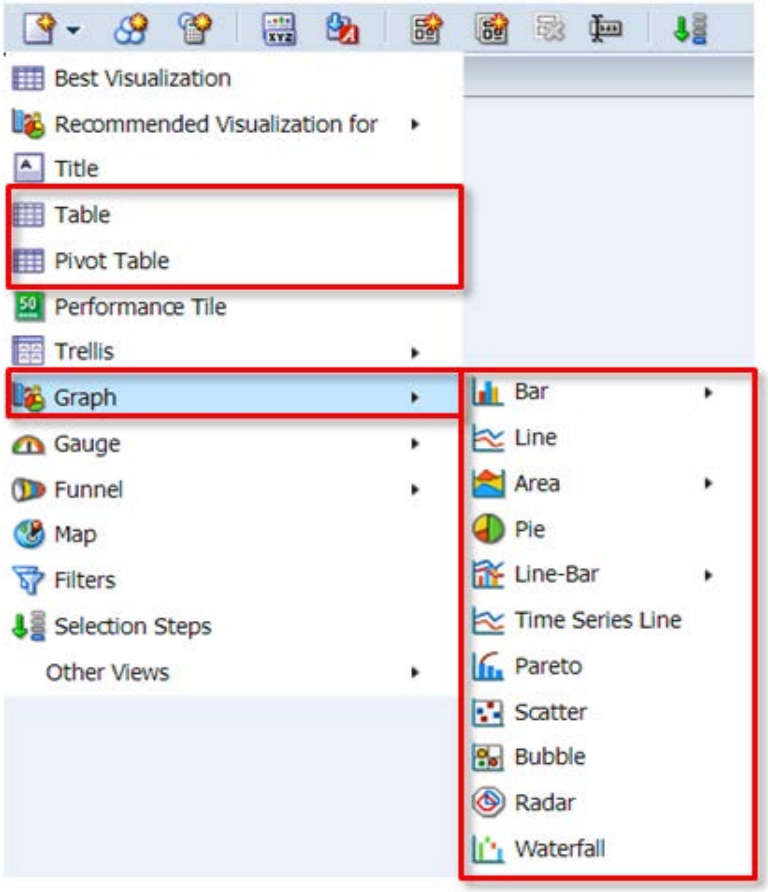
Sales Order Details by Sales Rep
Time run: 4/24/2013 10:34:18 PM

Vision Sweden; Vision UK and Ireland

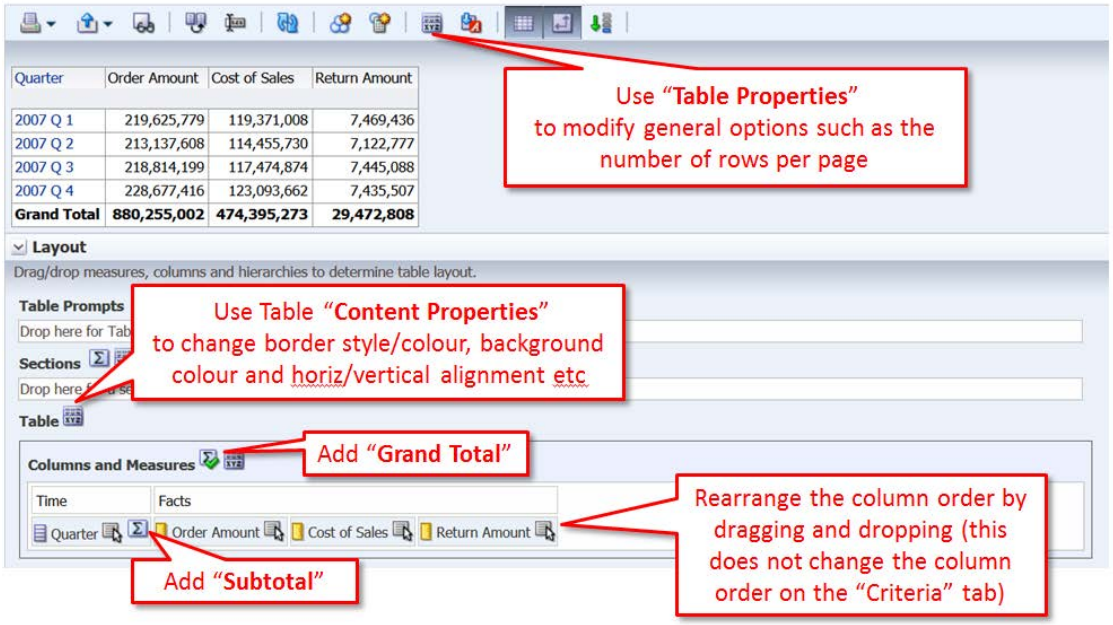
Sales Rep	Customer Name	Account Number	Order Number	Deal Size	Order Date	Order Type	Order Status	Product Category	Order Amount	Return Amount
Jackson, John	A. C. Networks	1143	123775	100K+	22-10-2007	MIXED	CLOSED	Capacitors	114,786	2,985
								Computer Systems	14,528	
								Desktop	14,528	
								Electronic Components	114,786	2,985
								Passives	114,786	2,985
								Vision High-Tech	129,314	2,985
	A. C. Networks Total								**502,727**	**11,939**
	ABC Corporation Worldwide	2575	123854	★★☆☆☆ 25K - 50K	28-12-2007	MIXED	CLOSED	Capacitors	38,455	2,157
								Electronic Components	38,455	2,157
								Passives	38,455	2,157
								Vision High-Tech	38,455	2,157
	ABC Corporation Worldwide Total								**153,821**	**8,628**
	Benson	2333	64543	100K+	22-10-2007	ORDER	CLOSED	Capacitors	136,874	4,288
								Electronic Components	136,874	4,288
								Passives	136,874	4,288
								Product Requirements	49,600	
								Vision High-Tech	186,474	4,288
	Benson Total								**646,696**	**17,151**

4.6. Tables, Pivot Tables and Graphs

OBIEE provides a variety of views to suit almost all reporting needs:



The Table view editor provides options to add Grand Totals and Subtotals as well as modifying the general layout and formatting. This helps us to fulfil the specified functional requirements.



The "Pivot Table" view is an extremely useful feature providing much greater flexibility that a standard "Table" view. Pivot tables have many features such as:

- Dimensions across two axis
- Calculated items
- Layers

	Total Revenue (Millions)			
	2007 Q 1	2007 Q 2	2007 Q 3	2007 Q 4
Business Group				
Vision Benelux	105.51	99.90	105.80	105.94
Vision East Europe	23.19	24.46	21.12	24.20
Vision Nordics	63.14	67.09	70.62	73.90
Vision UK and Ireland	4.20	3.87	2.62	5.73
Vision West Europe	16.11	10.70	11.20	11.47
Grand Total	**212.16**	**206.01**	**211.37**	**221.24**

Columns

Rows

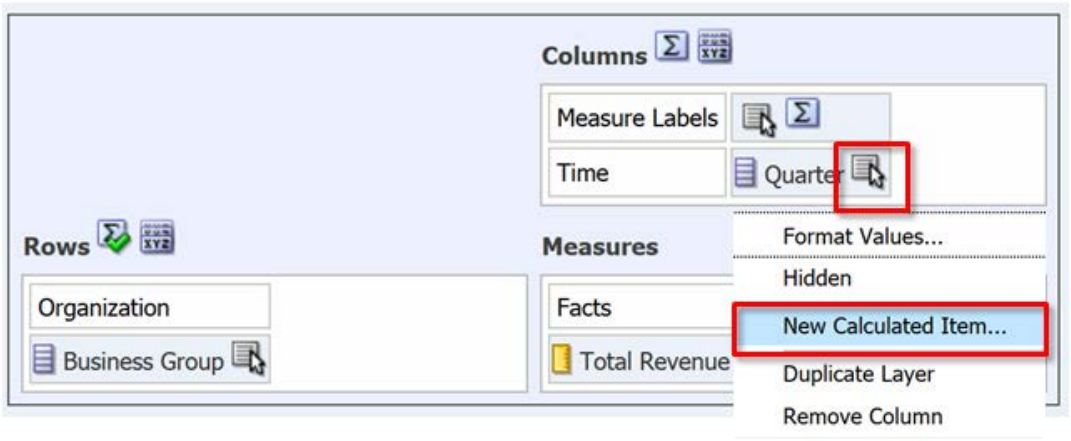
Measures

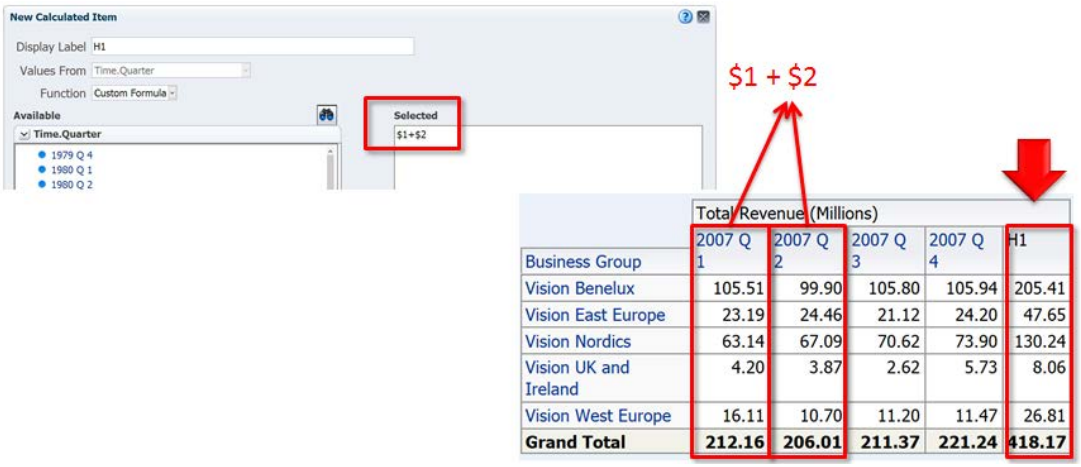
"Calculated Items" are a convenient way to add additional layers of calculation to the pivot tables. These additional calculations are performed once all the data has been fetched from the underlying database, so they do not impact database performance. In the case below, calculated items have been added to:

- Summarise the Quarterly data in to half-years "H1" and "H2"
- Categorise the Business Groups in to "West Europe" and "East Europe"

	# Orders					
Business Group	2007 Q 1	2007 Q 2	2007 Q 3	2007 Q 4	H1	H2
Vision Benelux	174	174	174	174	348	348
Vision East Europe	426	454	399	426	880	880
Vision Nordics	428	464	473	464	892	892
Vision UK and Ireland	78	80	52	104	158	158
Vision West Europe	192	174	182	174	366	366
West Europe	444	428	408	452	872	872
East Europe	854	918	872	890	1772	1772
Grand Total	**1298**	**1346**	**1280**	**1342**	**2644**	**2644**

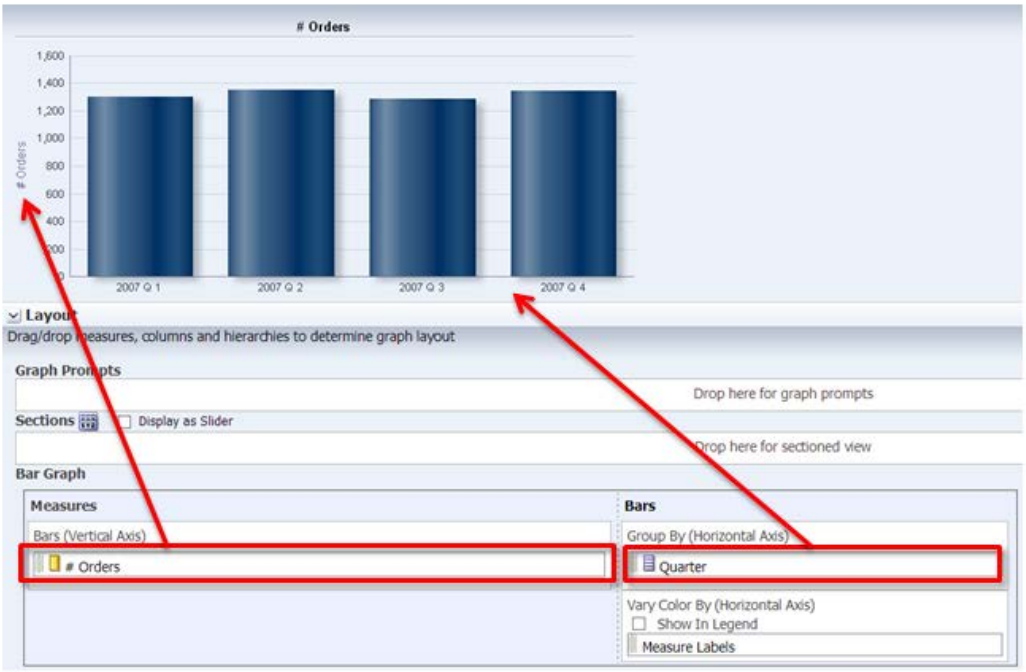
To do so:



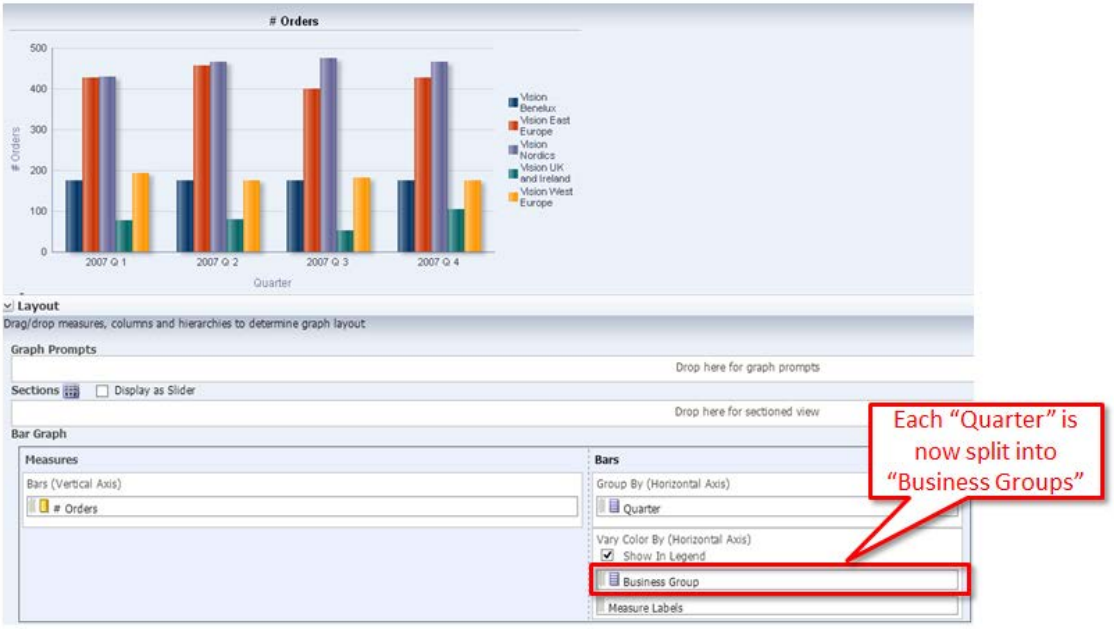


To fulfil the functional requirement regarding the graphics, OBIEE provides many types of Graph.

With Graphs, we start by choosing at least one Measure and also at least one Dimension to "group by":



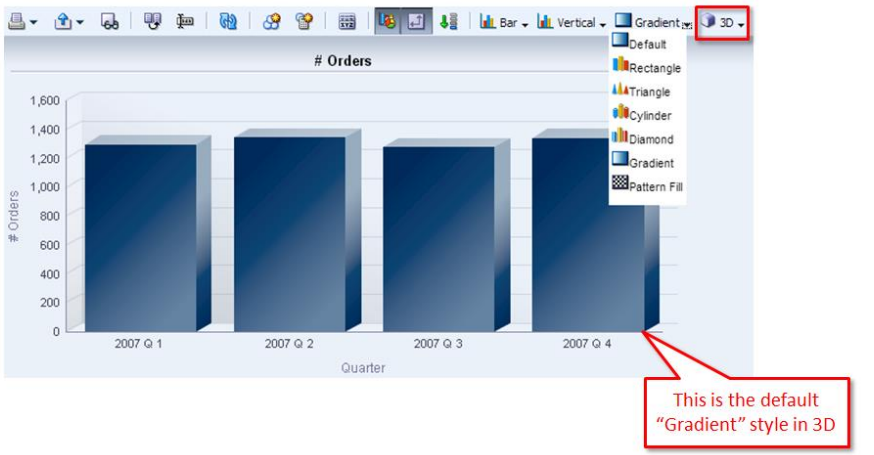
We can then also choose another Dimension to slice the data (using the "Vary Color By" option):



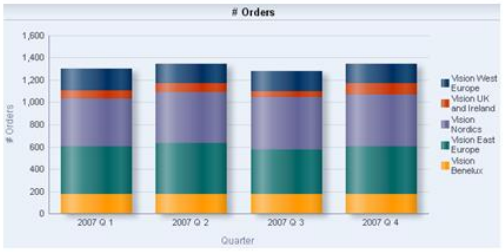
Graphs also come with a "**Display as Slider**" option.

Horizontal and Vertical "**Bar**" graphs are available:

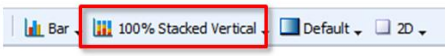
We also have an option to display the bars in 3D rendered with different shapes:

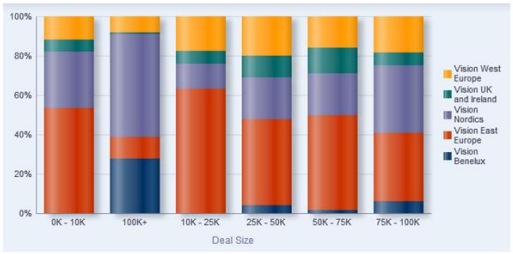


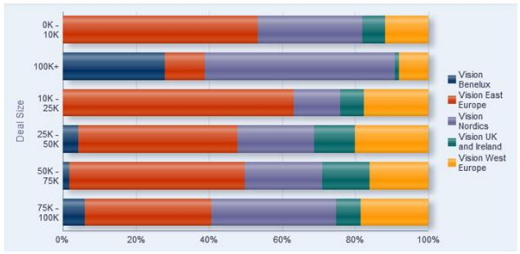
When we are varying the colour by a Dimension, we have the choice between “stacked” and “non-stacked”:



There are also “100% Stacked” variations which percentage weightings instead of their absolute values:



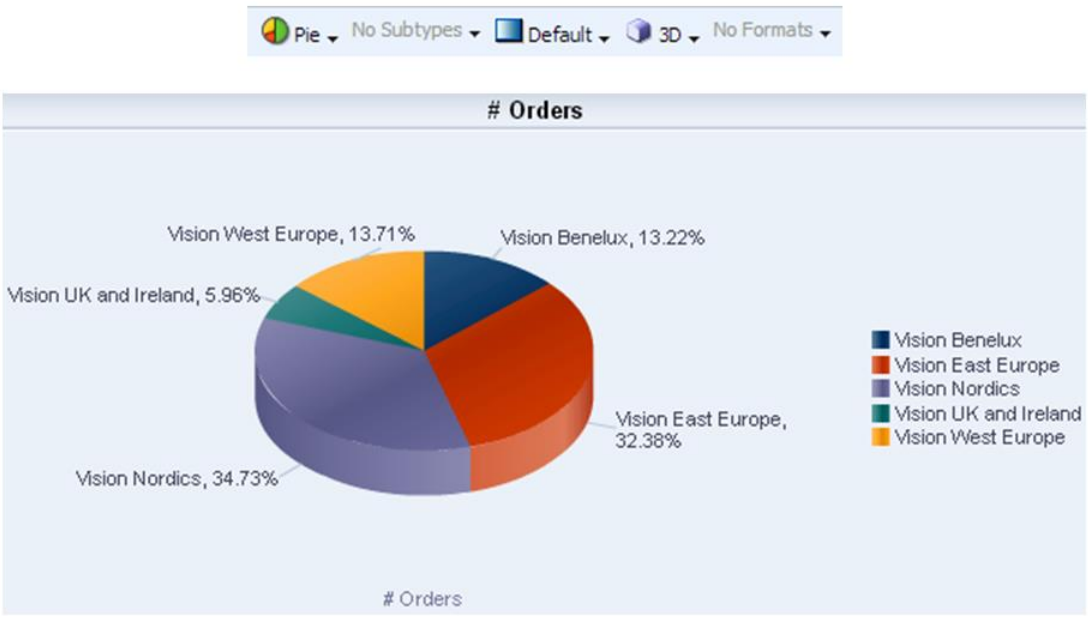




A "**Pie**" is another commonly used graph, available in 2D and 3D versions

Options allow us to:

- Display percentages or values
- Show just values or the name of the slice as well



Pie graphs also have a nice new feature where you can split up a single pie in to multiple ones based on a Dimension:

"**Line**" graphs are particularly useful for time-based charting. As with bar charts, we can display multiple measures on a line chart, or choose to have a number of lines based upon another dimension. Lines can be 2D or 3D.

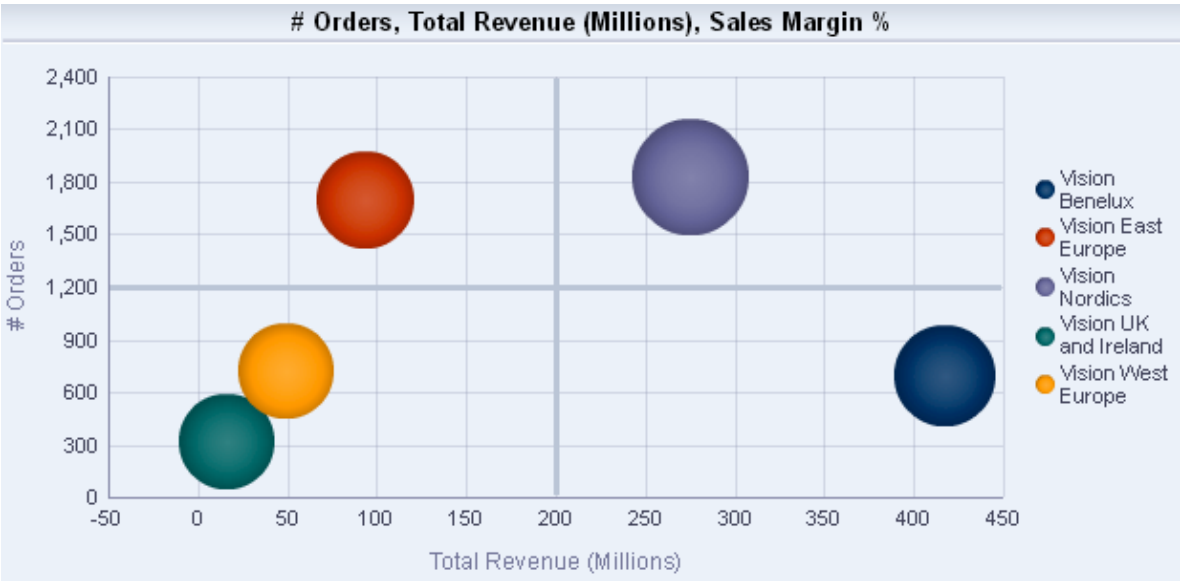
A "**Line Bar**" enables us to have one measure displayed as a line and another as a bar. There are two different X-axis, one on the left and the other on the right, both can have different scaling. In the example below, we have the monthly "Total Revenue" being displayed using bars, and the cumulative "YTD # Orders" being displayed as a line.

An "**Area**" graph is another form of line chart, showing clearly the different areas occupied by each dimension value on the chart – this is useful for seeing comparisons across time.

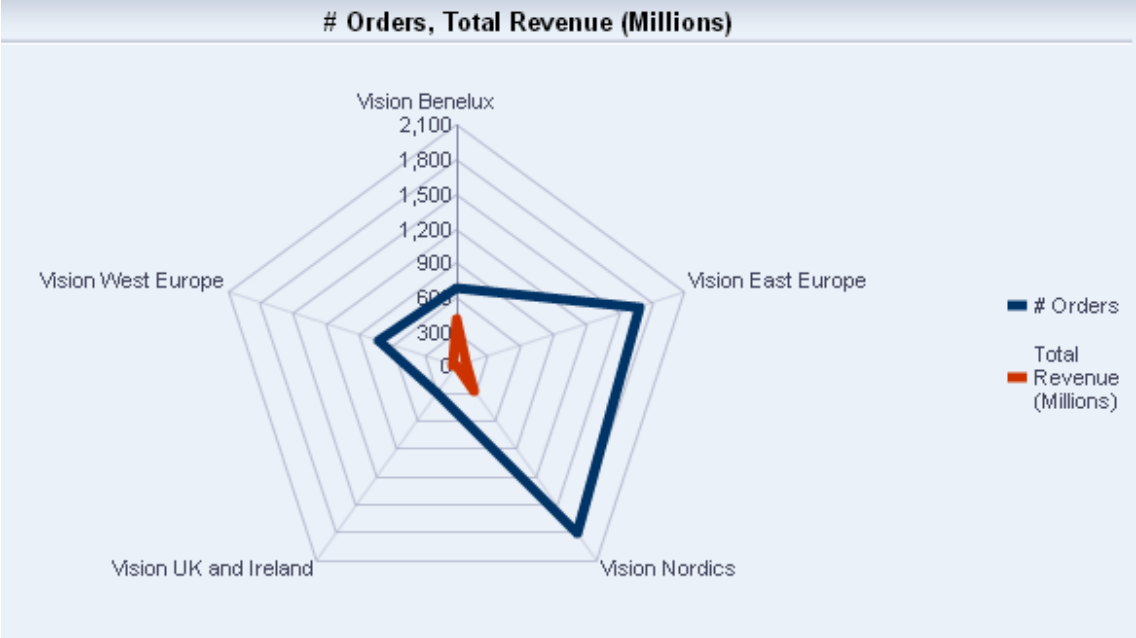
"**Bubble**" graphs provide a more advanced view where you can analyse 3 measures in one go.

In this example, we are comparing 3 measures for each "Business Group":

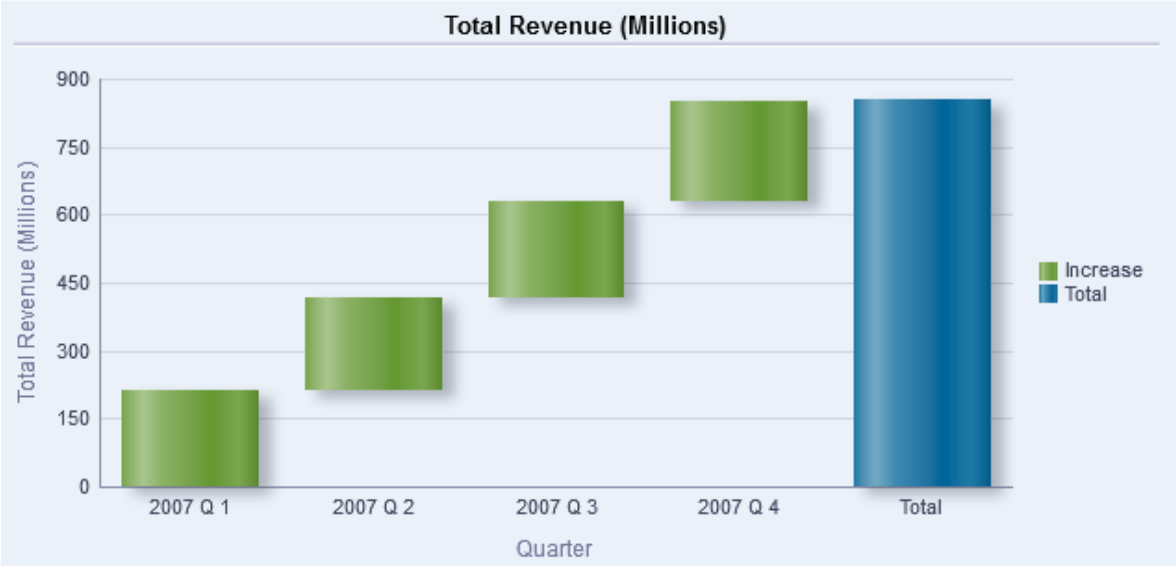
- # Orders *Y-Axis*
- Total Revenue *X-Axis*
- Sales Margin % *Size of Bubble*



"**Radar**" charts are great visual representations for data that can be organised by the hours of the clock (e.g. 0-12 or 0-24). But they are also useful to show how data is weighted towards certain Dimension values over others:

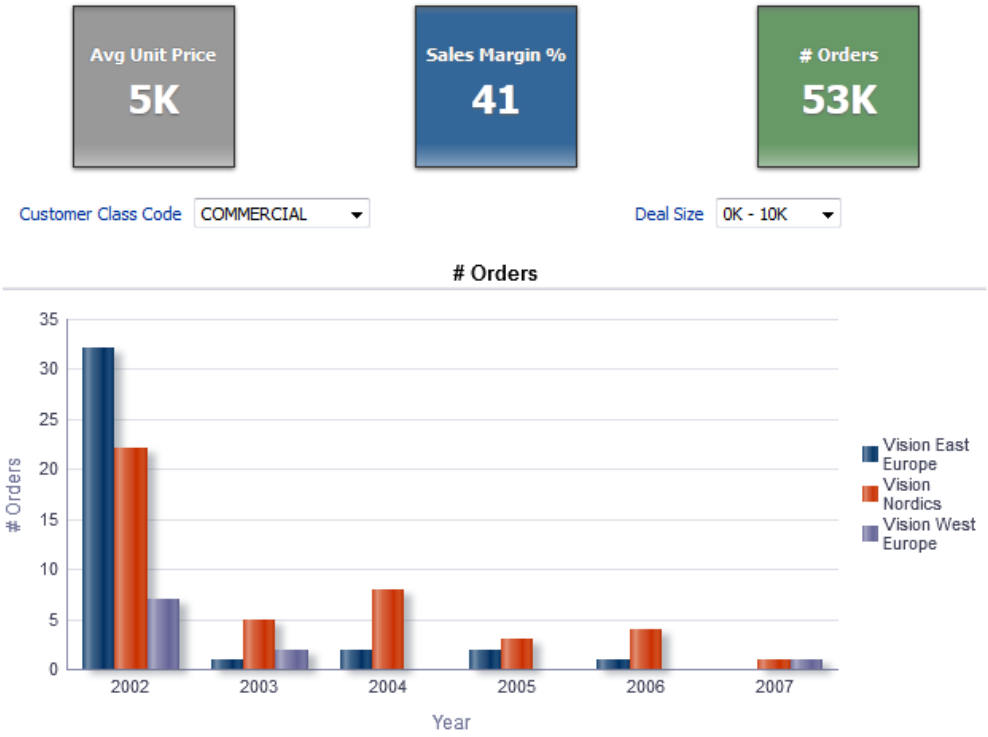


"**Waterfall**" charts are a type of floating-column chart. They show the cumulative increase or decrease from an initial value to a final value, with each column linked to the height of the previous column.



4.7. Other useful views

"Performance Tiles" provide simple but effective KPI-style indicators on the analyses. Each Performance Tile displays the "**grand total**" value for a specific metric.

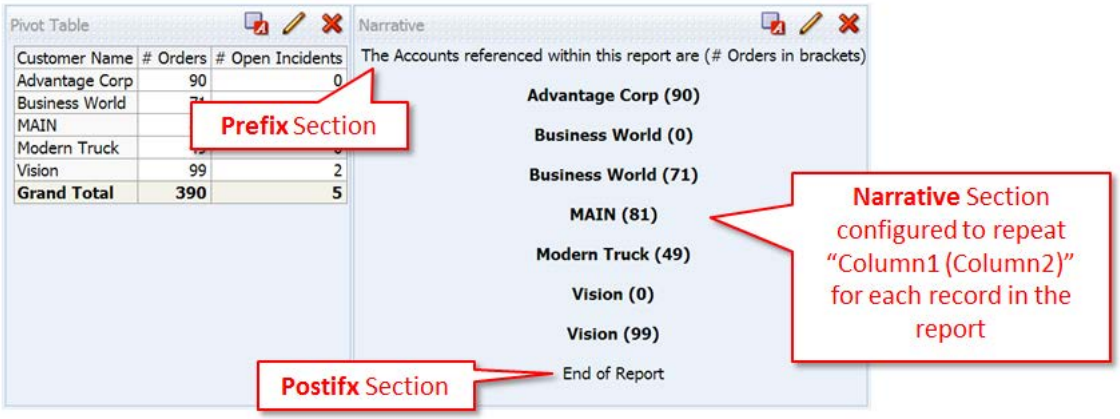


Business Group	Avg Unit Price	Sales Margin %	# Orders
Vision Benelux	10,143	38	7,181
Vision East Europe	1,034	36	15,299
Vision Nordics	1,140	61	19,104
Vision UK and Ireland	16,664	31	1,803
Vision West Europe	1,832	25	9,537
Grand Total	**4,705**	**41**	**52,924**

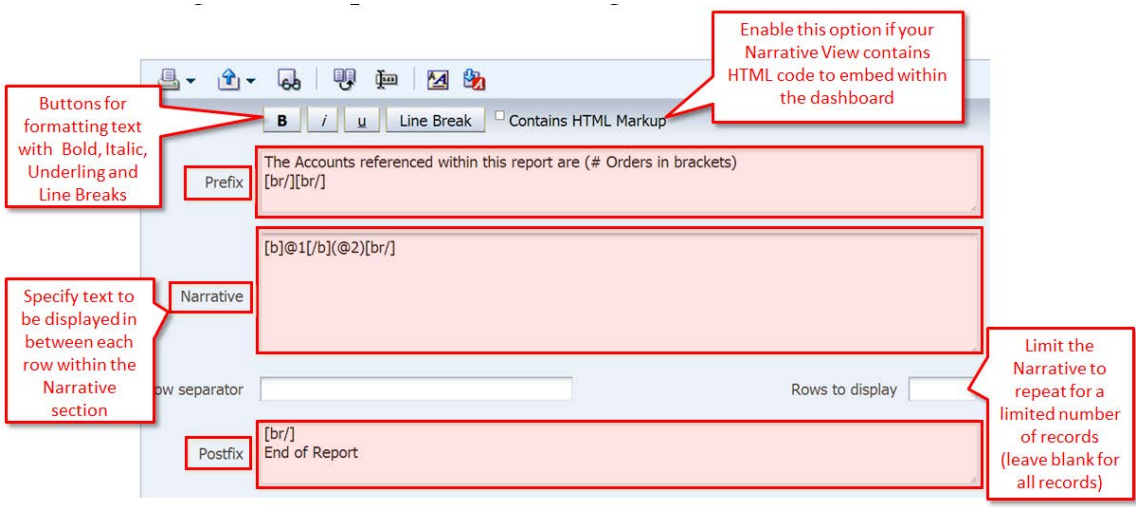
"Trellis Views" provide high-density visualisations with micro-charts:

Narrative Views are extremely versatile views, providing the flexibility to render results in an almost unlimited number of ways. A Narrative View is formed of 3 main sections:

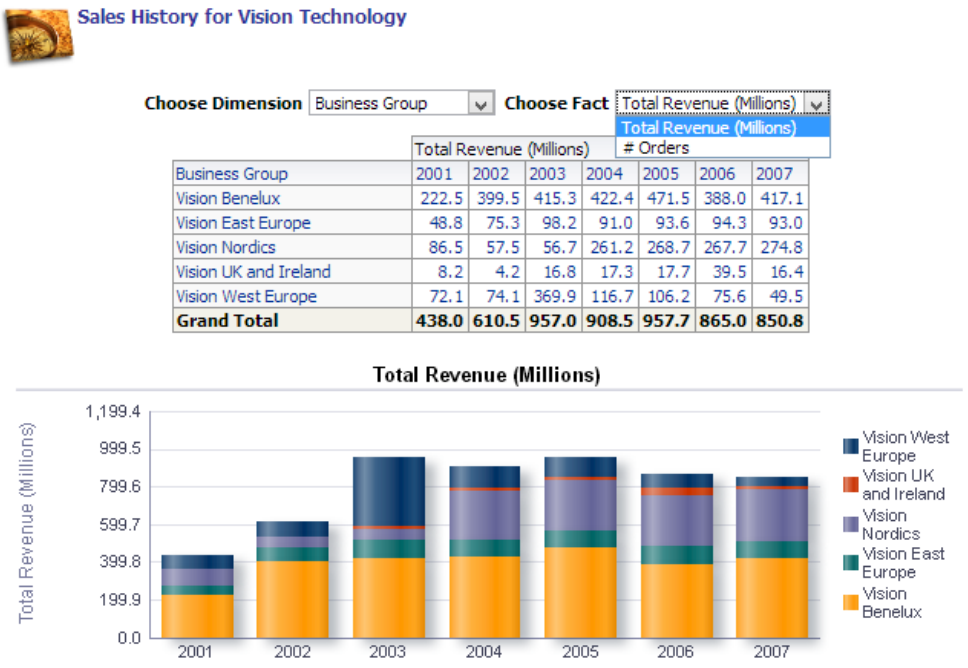
- Prefix - Text/formatting displayed once at the top
- Narrative - Text/formatting that repeats for every record
- Footer - Text/formatting displayed once at the bottom



Here is the editor for a Narrative View and this is how we configure it:



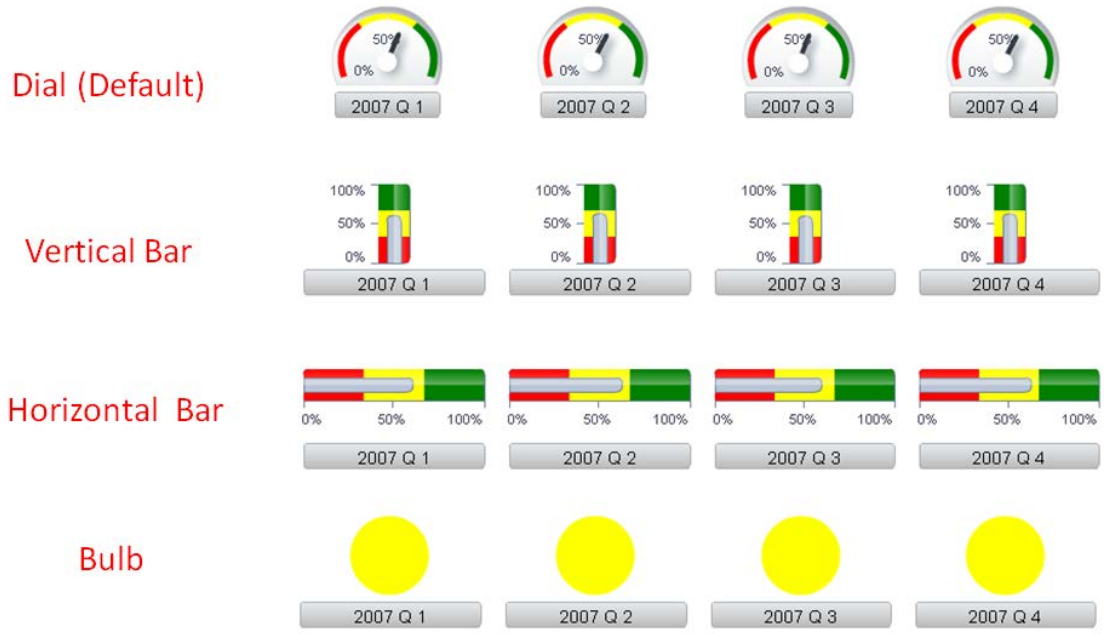
Column Selectors allow users to change the dimension and/or measure columns that are available on the request. This feature provides additional flexibility for the end users, allowing a single Analysis to behave as multiple Analyses. All views within the Analysis will refresh to use the column selected by the user.



View Selectors provide two key benefits:

- Users have the flexibility to choose a view that best suits their needs.
- The number of tables/graphs on display at any one time are greatly reduced, saving considerable space within the browser window.

Gauges are a great way to provide visualisations for KPIs and targets. There are four types of Gauge available:



4.8. Drill-Down and Navigation

Certain Dimension columns are configured to allow users to "drill-down" from one hierarchy level to another. A good example of this is with "Time":

Year	# Orders
2001	2686
2002	16232
2003	5241
2004	8743
2005	8016
2006	6740
2007	5266
Grand Total	**52924**

Year	Quarter	# Orders
2003	2003 Q 1	1268
	2003 Q 2	1266
	2003 Q 3	1299
	2003 Q 4	1408
Grand Total		**5241**

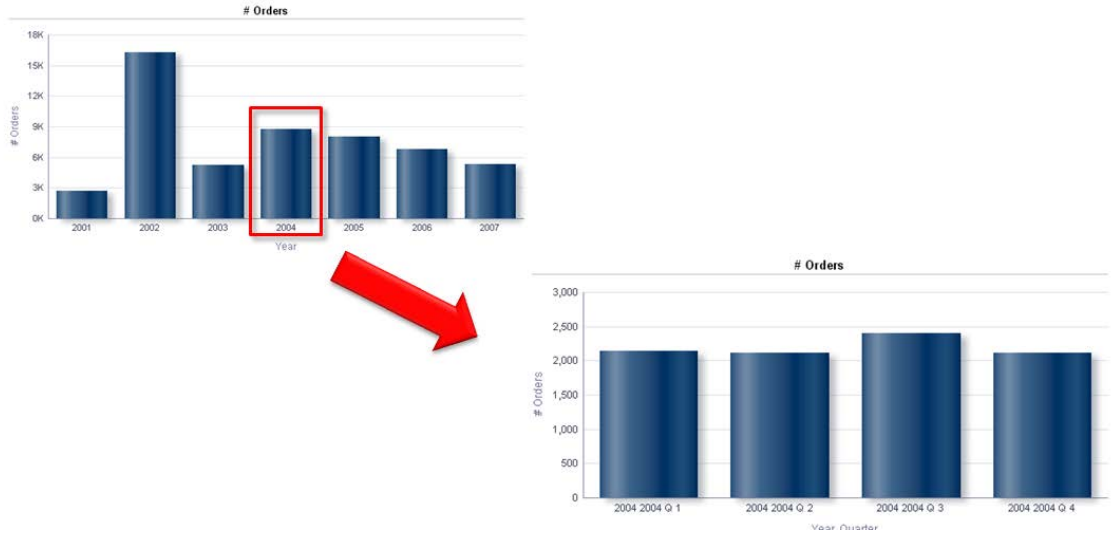
Year	Quarter	Month	# Orders
2003	2003 Q 3	2003 / 07	423
		2003 / 08	422
		2003 / 09	454
Grand Total			**1299**

We can also "drill-down" on the column heading, this will have the effect of drilling down to all the child values that occur at the next hierarchy level down:

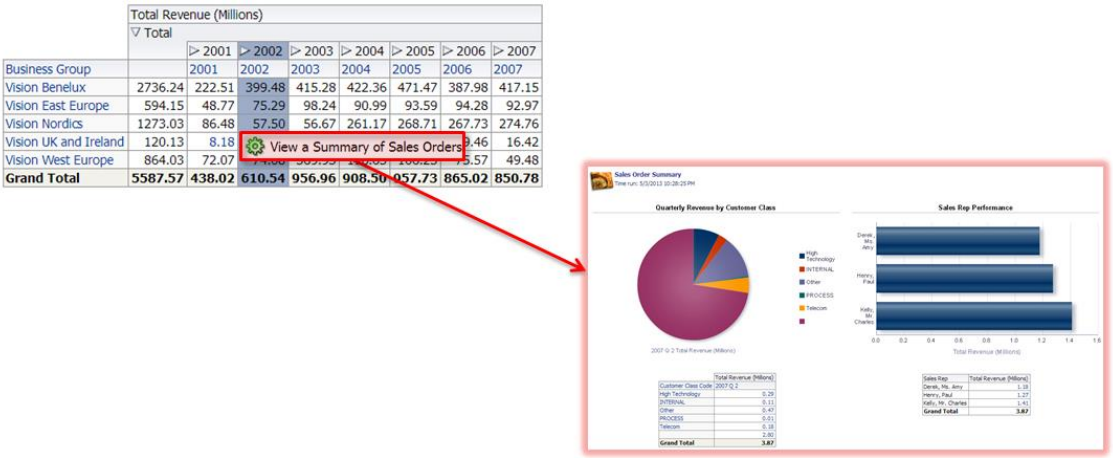
Year	# Orders
2001	2686
2002	16232
2003	5241
2004	8743
2005	8016
2006	6740
2007	5266
Grand Total	**52924**

Year	Quarter	# Orders
2001	2001 Q 1	1131
	2001 Q 2	1212
	2001 Q 3	28
	2001 Q 4	315
2002	2002 Q 1	1325
	2002 Q 2	7086
	2002 Q 3	6700
	2002 Q 4	1121
2003	2003 Q 1	1268
	2003 Q 2	1266
	2003 Q 3	1299
	2003 Q 4	1408
2004	2004 Q 1	2129
	2004 Q 2	2113
	2004 Q 3	2391
	2004 Q 4	2110
2005	2005 Q 1	2129
	2005 Q 2	1967
	2005 Q 3	1835
	2005 Q 4	2085
2006	2006 Q 1	2000
	2006 Q 2	1856
	2006 Q 3	1456
	2006 Q 4	1428
2007	2007 Q 1	1298
	2007 Q 2	1346

Drill-downs are automatically enabled on graphs by default too.



Oracle BI comes with a powerful feature called "**Action Framework**" which enables users to initiate various "Actions" from their Analyses/Dashboards. One of the Actions available is "**Navigate to BI Content**" which allows us to navigate directly from one Analysis to another Analysis or Dashboard Page.



4.9. Conditional Formatting

This OBIEE functionality helps us to fulfil another functional requirement. Conditional Formatting is a useful feature enabling us to highlight trends, exceptions and problems in your data. Conditional Formatting can be applied to:

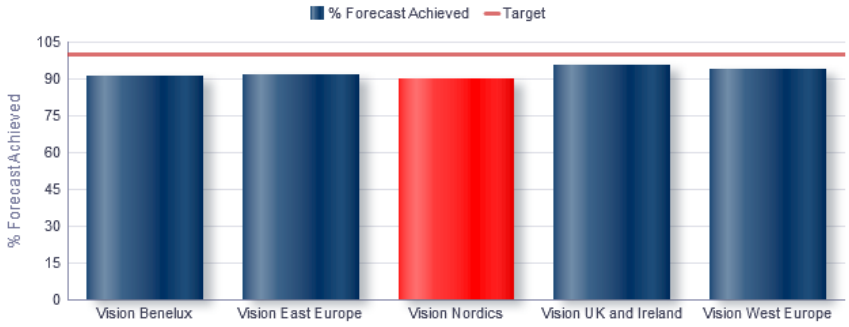
- Tables

Deal Size	Order Amount	Return Amount
0K - 10K	64,128,029	1,260,296
10K - 25K	106,140,562	1,998,369
25K - 50K	297,736,078	5,634,983
50K - 75K	452,048,223	8,220,082
75K - 100K	485,629,838	8,977,168
100K+	4,295,393,421	87,419,641
Grand Total	**5,701,076,150**	**113,510,539**

- Pivot Tables

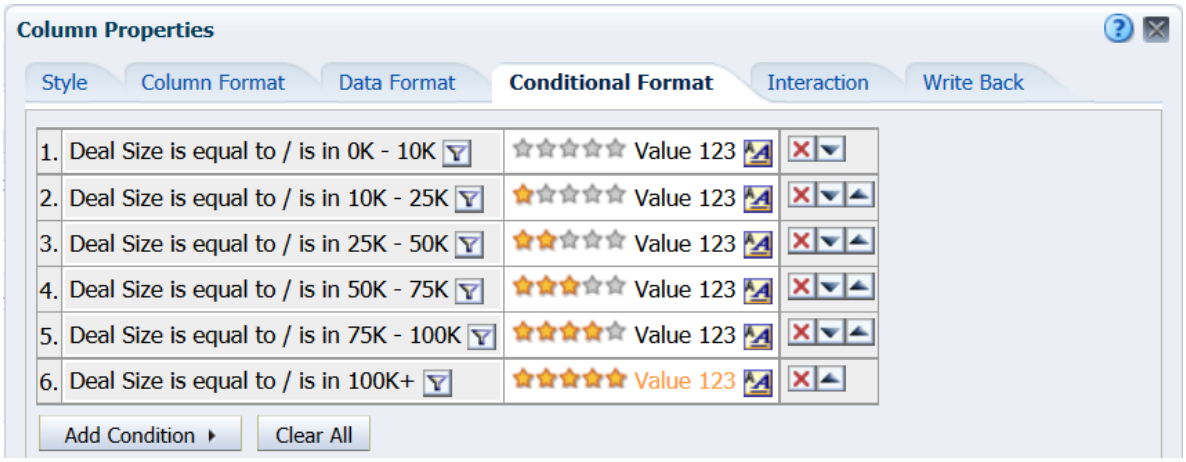
	% Forecast Achieved			
Organization	2007 Q 1	2007 Q 2	2007 Q 3	2007 Q 4
Total	90.72	90.77	93.31	94.75
Vision Benelux	90.55	88.04	89.93	93.63
Vision East Europe	91.55	103.07	96.21	101.45
Vision Nordics	89.81	90.70	97.42	91.69
Vision Finland	90.83	95.52	102.68	91.24
Vision Sweden	84.76	76.57	82.77	92.92
Vision UK and Ireland	95.15	88.49	91.21	92.46
Vision West Europe	93.24	93.64	97.00	118.15

- Graphs



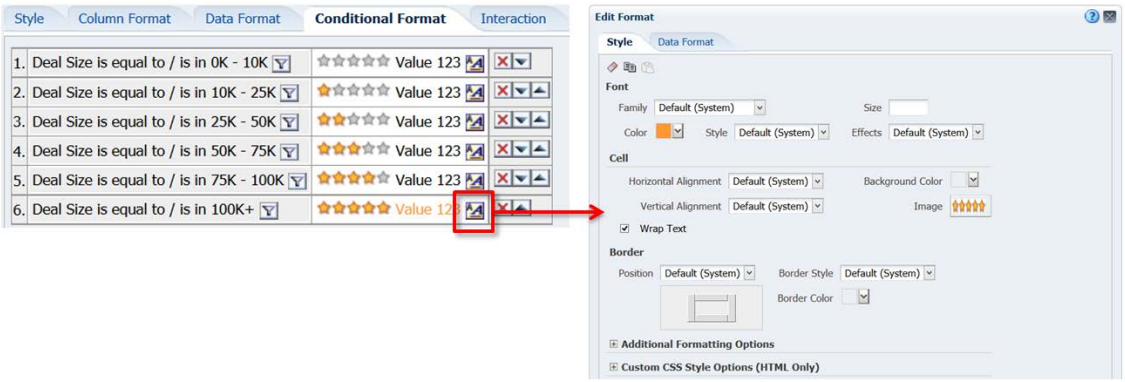
Conditional formatting is like a series of "If Then" statements:

- If [Condition 1] THEN [apply formatting 1]
- If [Condition 2] THEN [apply formatting 2]
- If [Condition 3] THEN [apply formatting 3]

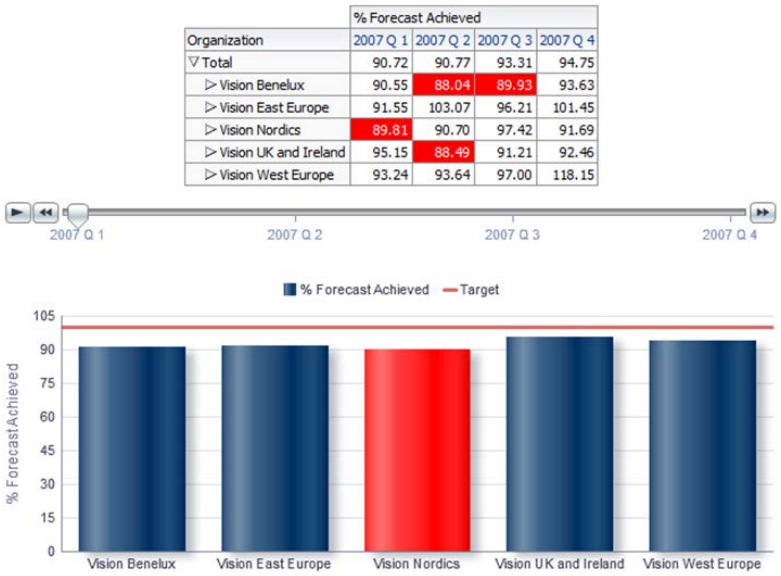


There is a whole range of formatting that can be applied conditionally:

- Font size, colour, style
- Cell alignment, background colour, image
- Border position, style, colour



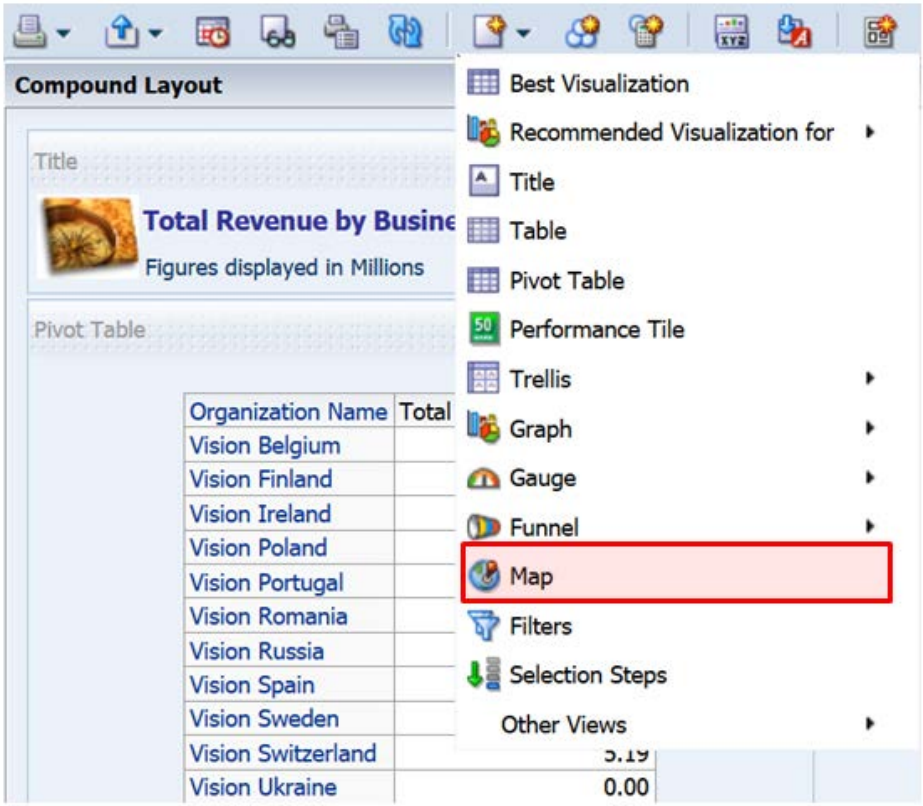
Another example:



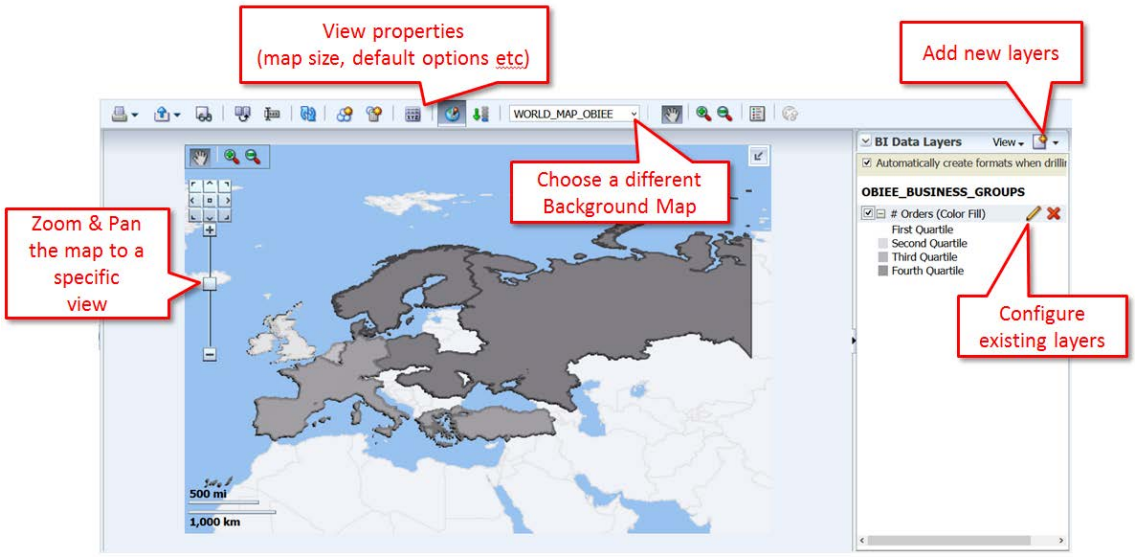
4.10. Embedded maps

The ability to embed maps in to Oracle BI Dashboards is an exciting feature available with OBIEE 11g. With this we also accomplish another functional requirement.

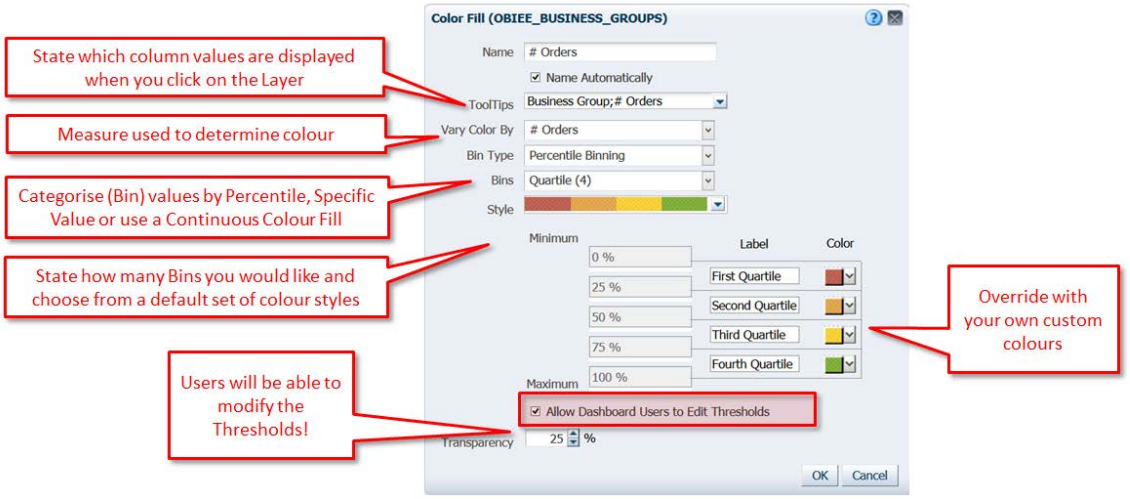
To create a map view we just need to make sure at least one of the columns can be rendered on a map. In this example, each "Organization Name" corresponds to a geographical country and this has been configured for display on a Map.



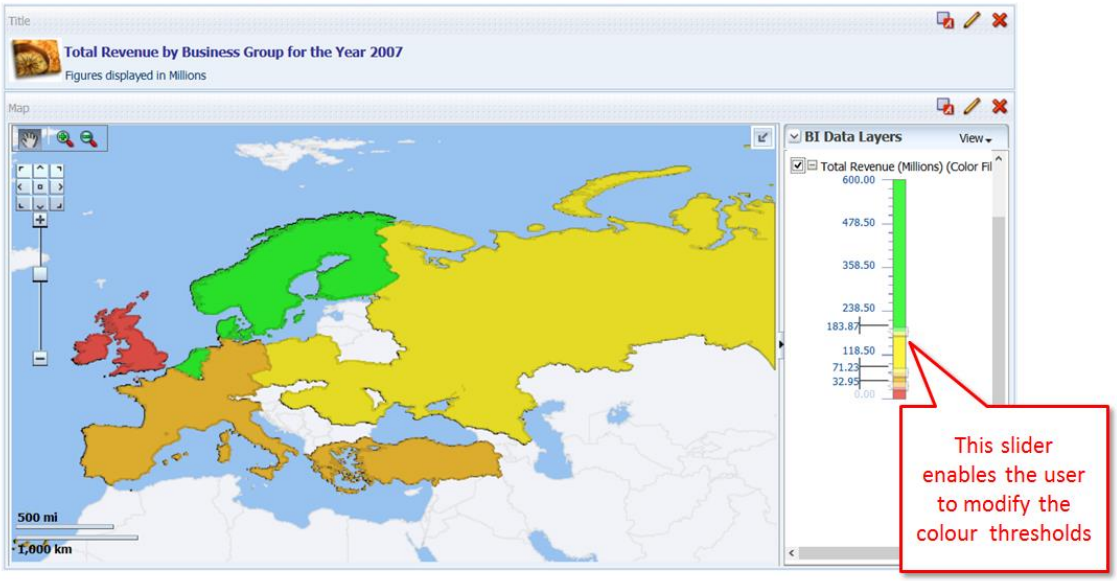
The map editor provides a number of options:



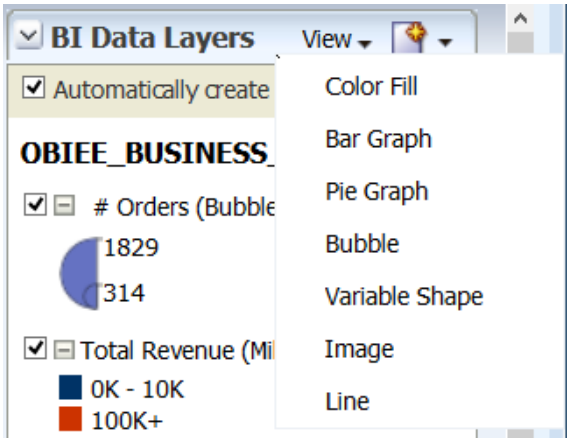
There is a lot of flexibility when it comes to formatting the Layers. In this example we have divided "# Orders" in to four different Bins and chosen our own custom colours.

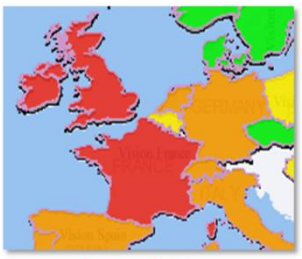


And here is the map now that its layer has been configured:



There are several different types of Layer you can add:



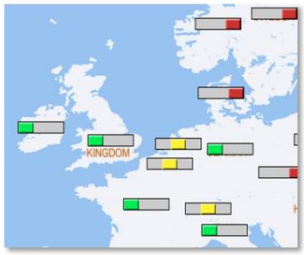

Colour Fill


Bar Graph

Pie Graph

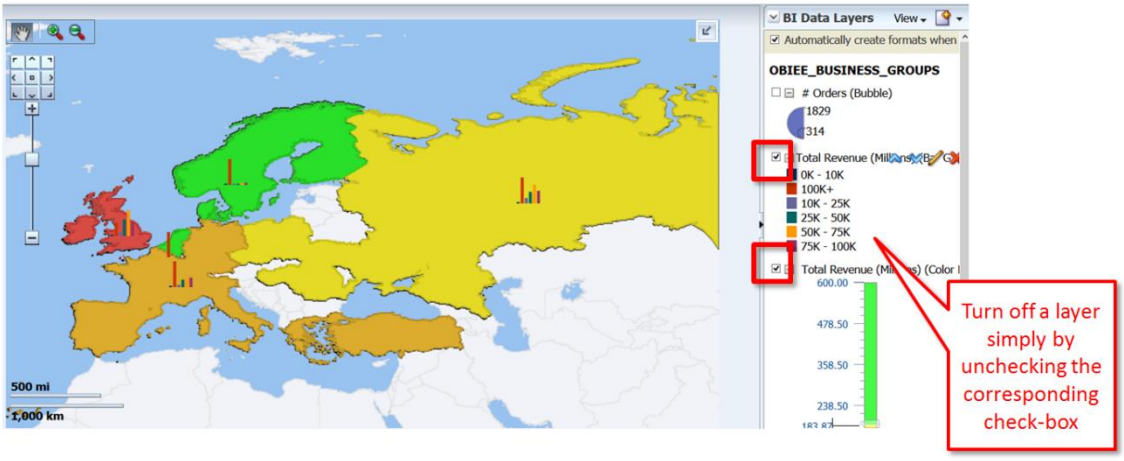
Bubble

Variable Shape



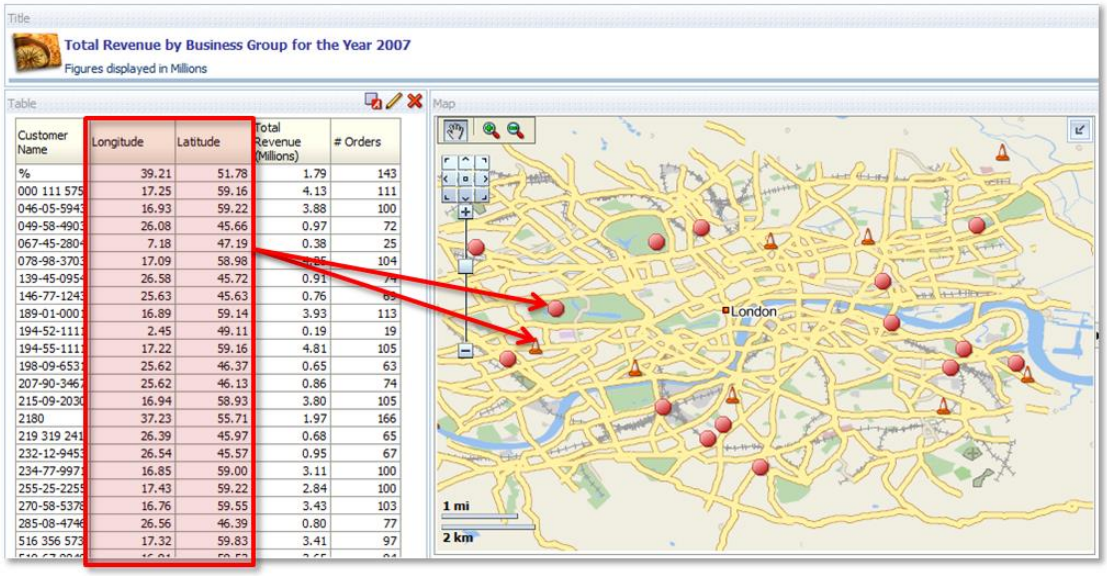
Image

The user can choose which layers to display on the map:

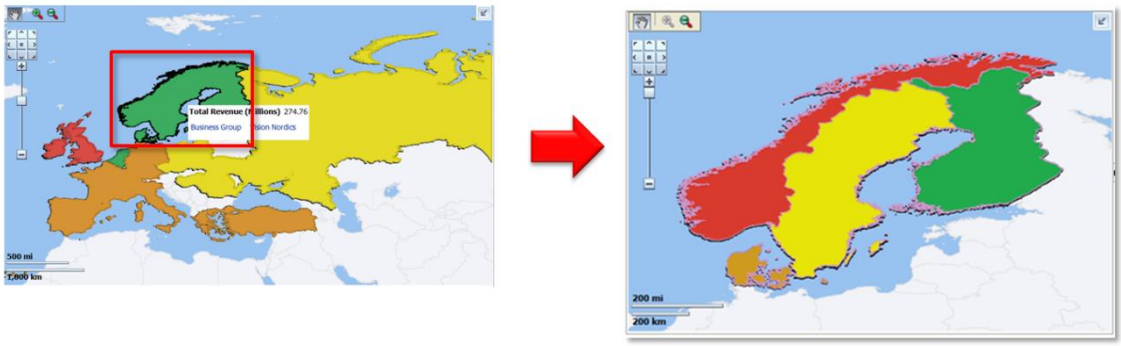


"Custom Point Layers" are where the co-ordinates are taken from the Analysis itself:

- Mapping coordinates can come from any type of data source!
- Longitude/Latitude co-ordinates can be provided in a single column or separate columns
- Supported with Bubble, Variable Shape and Image layers



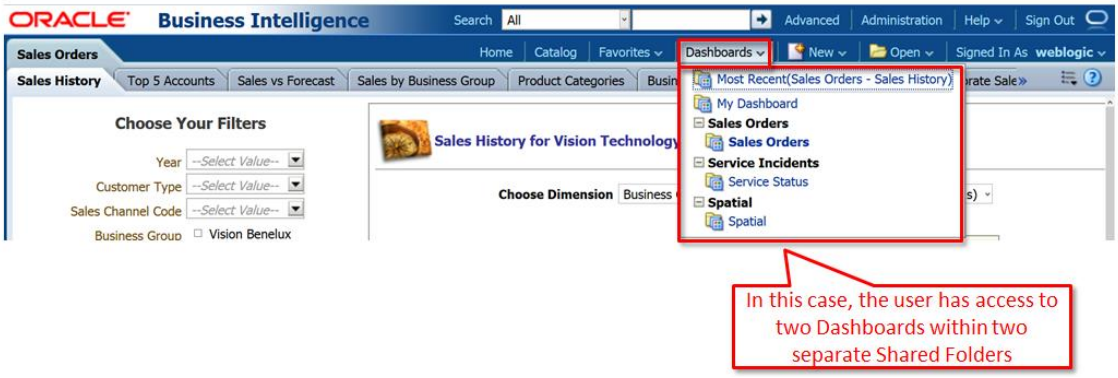
Drill-downs work “out-of-the-box” on embedded maps just like they do on Tables, Pivot Tables and Graphs. In this example, we drilled-down on the “Vision Nordics” Business Group. The map automatically changed to show data by Organization for the “Vision Nordics” Business Group. Note that the map zoomed in automatically.



4.11. Dashboards

To meet the final functional requirement we have OBIEE Dashboards. Most Oracle BI end users will consume their reporting content via one or more interactive "Dashboards". Dashboards can be role-based and personalised.

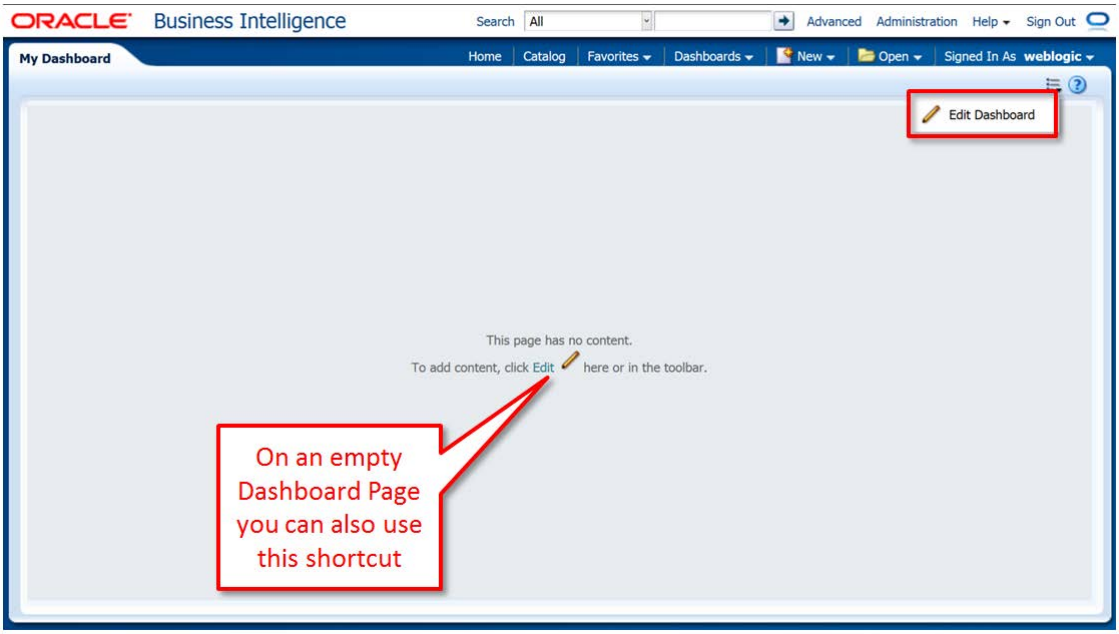
Dashboards can be accessed at any time using the "Dashboards" menu at the top of the screen.



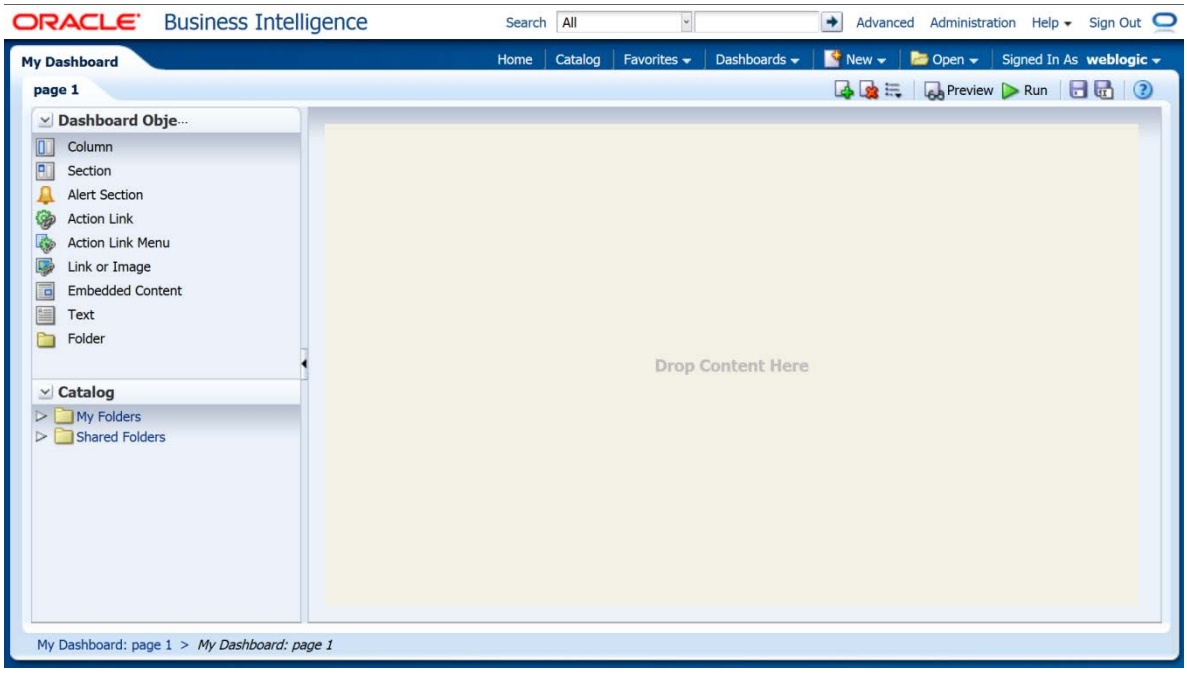
A Dashboard consists of:

- One or more Pages
- A Dashboard Prompt (to allow users to apply their own filters)
- One or more Analyses

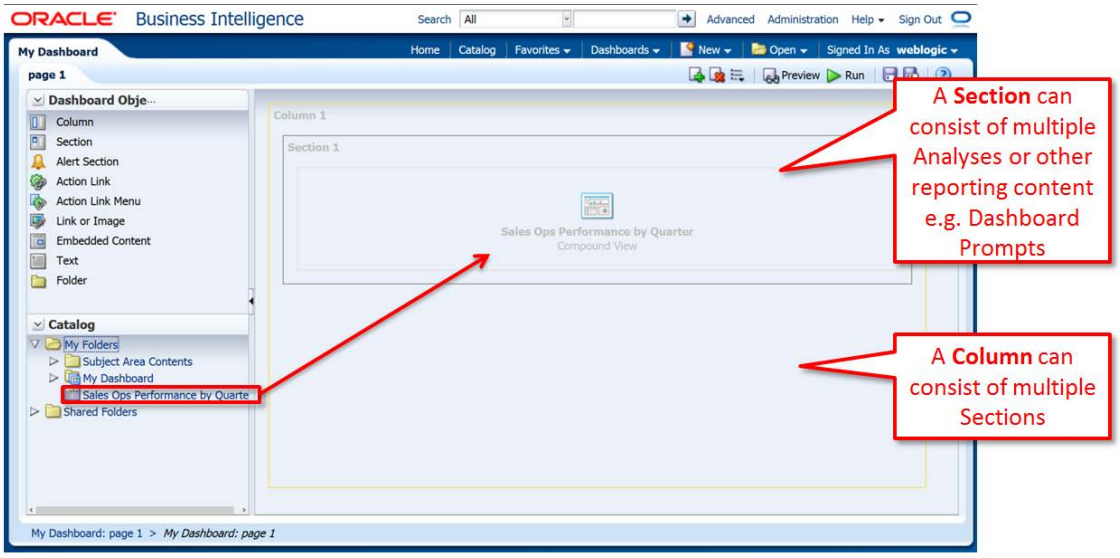
To edit a dashboard go to the "Page Options > Edit Dashboard" menu on the right-hand side.



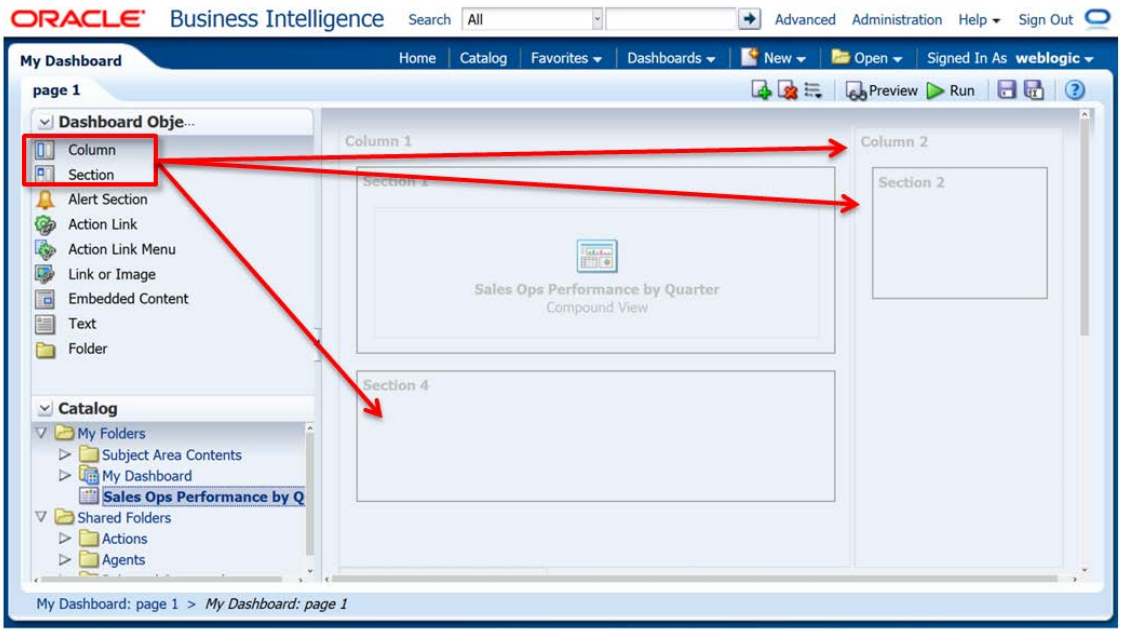
The "Dashboard Editor" will then open up. By default, we will have a single empty Dashboard Page called "page 1".



We can start adding content by dragging objects from the "Catalog" window pane. The object will be placed inside a new "**Section**" and "**Column**".

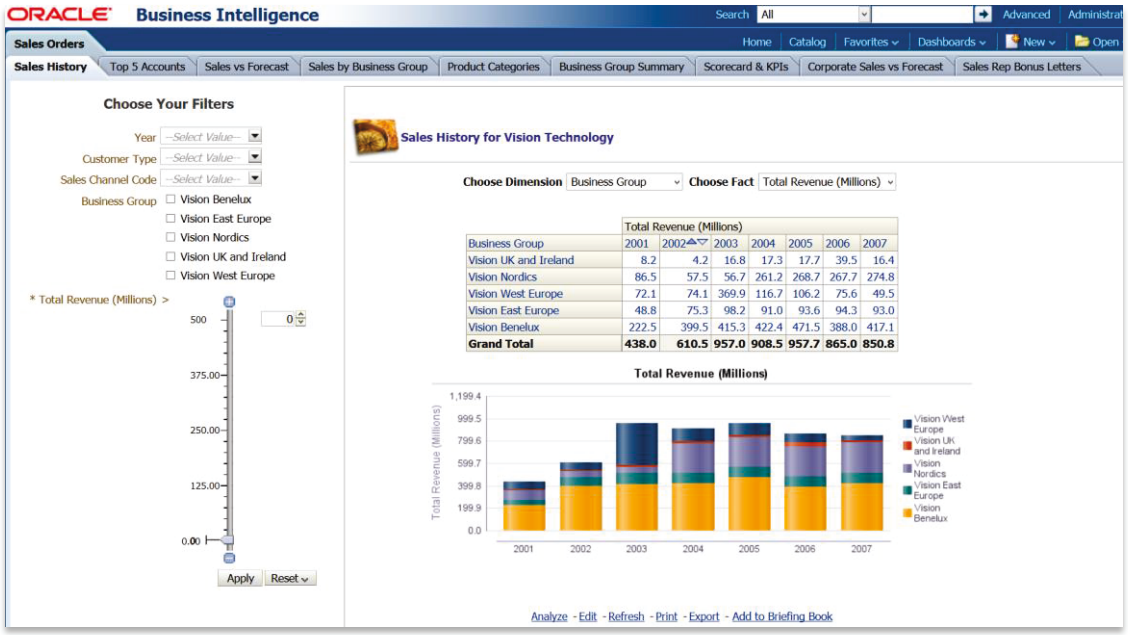


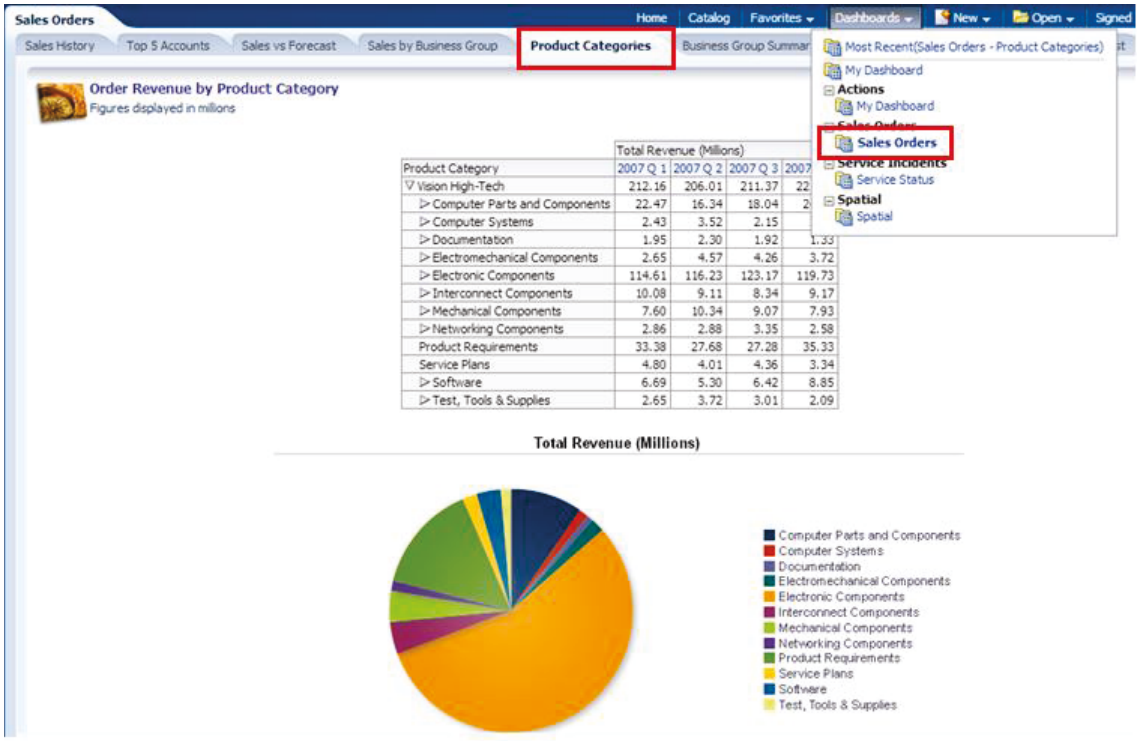
We drag additional Sections and Columns from the "Dashboard Objects" panel, place Sections above or below existing Sections or place Column above, below, left or right of existing Columns.

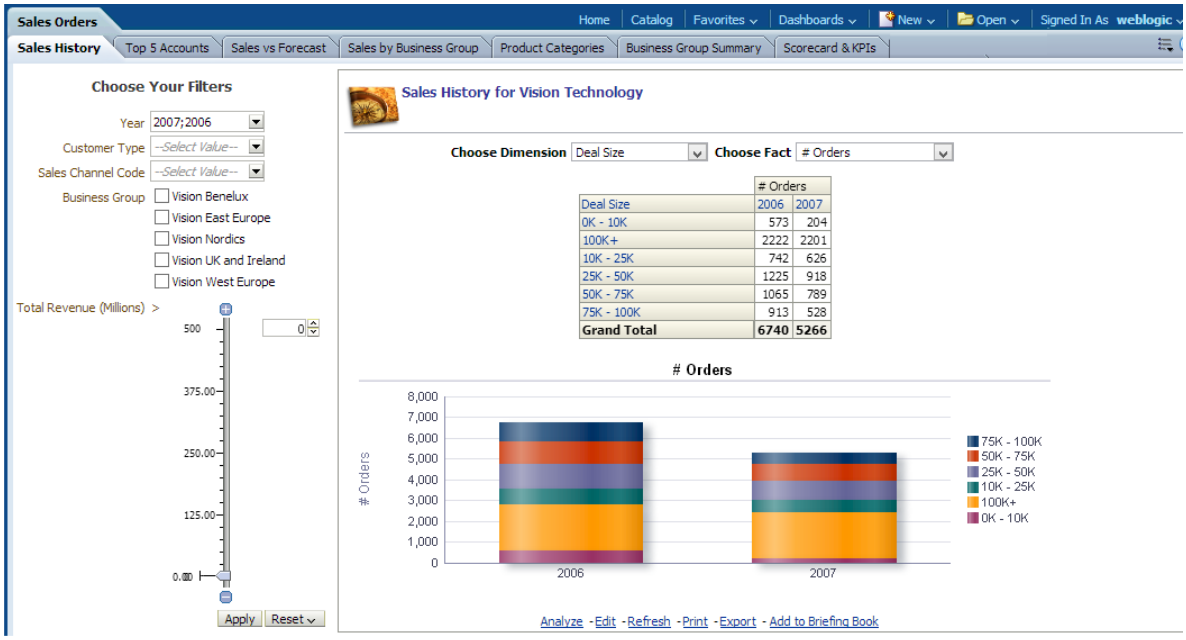


4.12. The end result: A fully prepared Dashboard

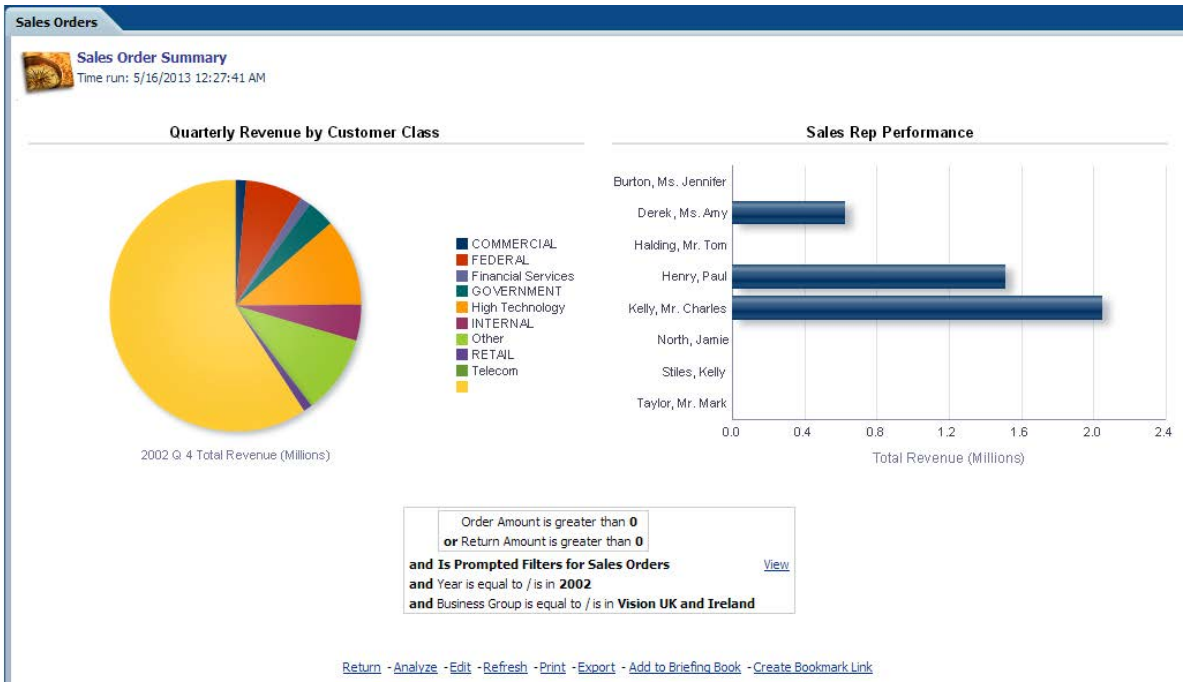
After all the process explained before of creating and adding contents such as reports or other element, the completed and fully prepared dashboard to help important decision making will look like this:







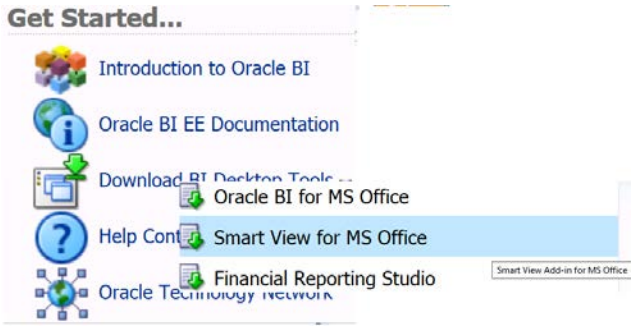
If you navigate from one to another Analysis from a Dashboard Page, you will see:



5. Future Projects or Improvements

In this chapter I will explain and suggest possible improvements for the present TFG and related possibilities for other TFG.

- **Implementation of Smart View for MS Office in the OBIEE Application**: Smart View is a component used for integration with Microsoft Office (Power Point, Word, Excel). A light weight client install is necessary, but it can easily be done from the Dashboard Home Page.



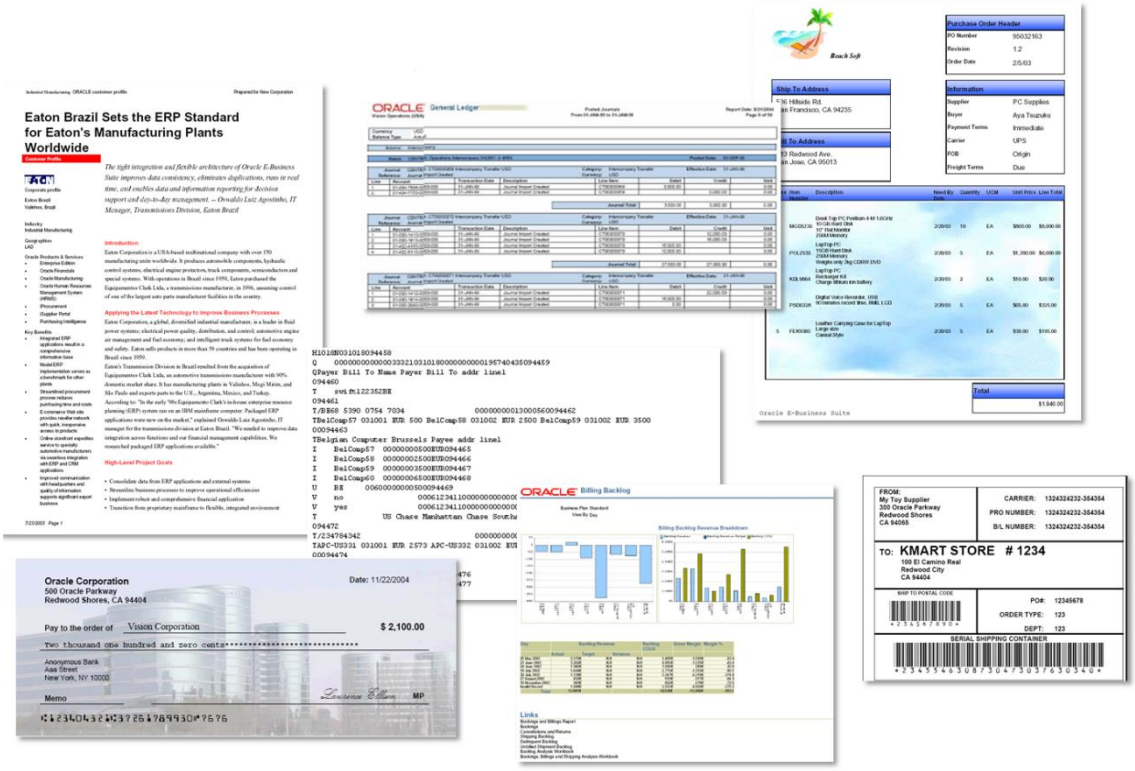
After installation is complete, it will be possible to export elements from our OBIEE to any Office format. MS Office is the most used software for documentation worldwide so this improvement would have a great impact.

- **Implement the Delivers functionality in OBIEE**: Delivers is the component of Oracle BI which allows you to automate different event-driven actions such as alerting, scheduled report distribution and conditional action execution. For example, if the sales in a region are dropping, send an alert and a report or an email to the person responsible in that region.

- **Create high fidelity reports with Oracle Business Intelligence Publisher:** Although our reports and dashboards in OBIEE have a great format quality and are very good presented, there is a certain limit to the freedom you have when designing the format. This can be overcome with Oracle Business Intelligence Publisher which is an enterprise reporting solution to author, manage, and deliver all types of highly formatted documents ("pixel perfect" reporting). It is built on open standards, so users can create reports to run against practically any data source and use BI Publisher APIs to build custom applications leveraging existing data sources and infrastructure. End users can easily design report layouts using familiar desktop tools, which improves productivity. In addition, Publisher is extremely efficient and highly scalable. It can also

generate tens of thousands of documents per hour with minimal impact to transactional systems. Reports can be viewed online or scheduled for delivery to a wide range of destinations.

Here are some examples of some high fidelity reports created with Publisher:



6. Conclusions

After finishing the development of this TFG- project with a research of BI technologies, OBIEE and the creation of the proof of concept, it is time to draw a conclusion from it. As it was stated in the introduction section, the main purpose of this project was to research, explain and promote OBIEE in the academic and professional environment.

Oracle Database, Data Warehousing and other important technologies in which OBIEE is based have been researched and analyzed. Then I have shown the global architecture of OBIEE, its components, and how it works so that any developer with a minimum of technical knowledge is able to globally understand OBIEE.

In the proof of concept we have listed complex requirements that are normally difficult to implement. Nevertheless, in the same proof of concept we have modelled and designed an OBIEE web application that easily and efficiently fulfills all the requirements. In the process we have shown the great potential that OBIEE has for big enterprises: interpreting voluminous data friendly. All of this will help the enterprise to create an effective strategy that can provide a competitive market advantage and long-term stability.

After reading this project, the reader should have a great knowledge about OBIEE and how BI is in an average international company. In addition, providing that an OBIEE environment is given to the reader, he could even model an RPD, create a Subject Area and from there design his own report to include in an OBIEE Dashboard.

OBIEE is one of the best BI tool in the world that helps thousands of companies improve their business. At the same time this companies need millions of specialized BI consultants and qualified personnel to implement OBIEE in their business. This generates billions of euros for both big international enterprises and consulting companies.

As shown in this TFG OBIEE is very easy to learn and use, has a great potential to create very interesting web applications. This makes OBIEE a very good option for any recently graduated computer engineer. I have been working as a BI consultant specialized in OBIEE for approximately two years and I completely recommend it to anyone.

Therefore, we can say that the objective of researching, showing and promoting OBIEE was successfully achieved.

7. Bibliography:

- *Oracle. "Oracle Database. Data Warehousing Guide". 11g release 2 (11.2). E25554-02. [Viewed: March 2014]. Available at http://docs.oracle.com/cd/E11882_01/server.112/e25554.pdf*

- *Cuneyt Yilmaz. Oracle Business Intelligence 11g R1 Cookbook. Packt Publishing(Publisher). 2013. [Viewed: April 2014]. ISBN: 978-1-84968-600-6. Available at: http://www.it-ebooks.info/book/3018/*

- *Rick Greenwald, Robert Stackowiak and Jonathan Stern. Oracle Essentials, 5th Edition. O'Reilly Media (Publisher). 2013. [Viewed: February 2014]. ISBN: 978-1-4493-4303-3. Available at: http://www.it-ebooks.info/book/3058/*

- *Sam R. Alapati and Charles Kim. Oracle Database 11g. Apress (Publisher). 2007. [Viewed: May 2014]. ISBN: 978-1-59059-910-5. Available at http://www.it-ebooks.info/book/1839/*

- *Oracle. Oracle documentation and other resources. 2014. [Viewed May 2014]. Available at http://www.oracle.com/*

- *Encyclopaedia Britannica. Britannica.com. 2014. [Viewed June 2014]. Available at http://www.britannica.com/*

8. Appendix

8.1. Appendix 1: Budget Estimation

In order to make the budget estimation for this project, the following elements have been taken into account:

Note: all the currencies in the cost field are in Euros.

Hardware:

Element	Cost per unit	number of units	Total Cost
PC for development	400	6	2400
Machine to install OBIEE	2000	2	4000
Total			6400

For the PCs for development the cost refers to the renting price for the PC during the project period.

Software:

Element	Cost per unit	number of units	Total Cost
Windows 8 License	160	3	480
Office License	200	3	600
Oracle Business Intelligence Enterprise Edition	15400	1	15400
Total			16480

Personnel costs:

We consider that a working day consists of 7.5 hours.

Person	Cost per working day	number of days	Total Cost
Consultant	400	50	20000
Specialist	600	40	24000
Architect	800	26	20800
Total			64800

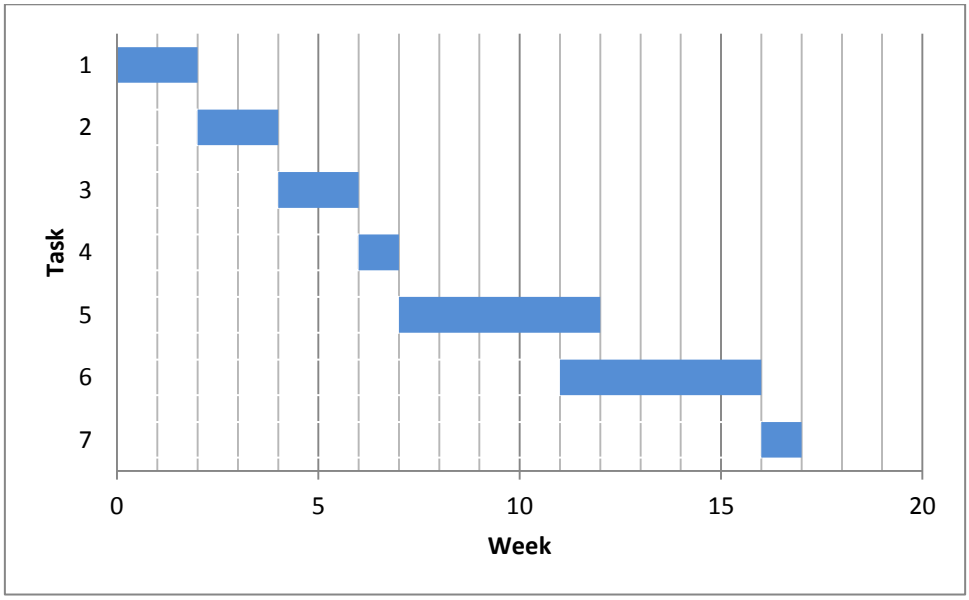
Total Cost:

Element	Cost
Hardware Cost	6400
Software Cost	16480
Personnel Cost	64800
Total Cost	87680

8.2. Task Planning

In this section we describe the task planning for the implementation of this project.

Task	Name	Invested time (days)	Starting Week	Finishing Week	Resource Name
1	Requirements garhering with client	10	1	2	Consultant
	Requirements garhering with client	4	2	2	Specialist
	Requirements garhering with client	4	2	2	Architect
2	Requirements definition: creation of documentation	10	3	4	Consultant
	Requirements definition: creation of documentation	5	3	3	Specialist
	Requirements definition: creation of documentation	2	4	4	Architect
3	Present information system and datasources investigation	10	5	6	Consultant
	Present information system and datasources investigation	6	5	6	Specialist
	Present information system and datasources investigation	4	5	5	Architect
4	OBIEE deployment and server configuration	3	7	7	Consultant
	OBIEE deployment and server configuration	3	7	7	Specialist
	OBIEE deployment and server configuration	3	7	7	Architect
5	Data arquitecture and modelling	20	7	8	Specialist
	Data arquitecture and modelling	10	12	12	Architect
6	Reports and Dashboards design	16	11	16	Consultant
	Reports and Dashboards design	12	13	16	Specialist
7	Quality ascertainments and global checks	3	17	17	Architect



8.3. Regulation Framework

As regards the regulation framework for the implementation of this project, the project will take place in Spain, and henceforth will abide by Spanish Law. The principal and most relevant law that affects this project is "the organic law 15/1999", which deal with personal data protection. The application of this law is compulsive. This project and the application developed will strictly abide by it.

All the information regarding this law can be found at Spanish Government website:

http://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf

Note: Here is the law that we will abide by. Since it is a Spanish law, it will be added directly in Spanish. This will be the only part of the TFG that is not in English. Translation of this law was taken into consideration, but was dismissed due to overextend the target of this project.

Artículo 1. Objeto.

La presente Ley Orgánica tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.

Artículo 2. Ámbito de aplicación

1. La presente Ley Orgánica será de aplicación a los datos de carácter personal registrados en soporte físico, que los haga susceptibles de tratamiento, y a toda modalidad de uso posterior de estos datos por los sectores público y privado.

Se regirá por la presente Ley Orgánica todo tratamiento de datos de carácter personal:

a. Cuando el tratamiento sea efectuado en territorio español en el marco de las actividades de un establecimiento del responsable del tratamiento.

b. Cuando al responsable del tratamiento no establecido en territorio español, le sea de aplicación la legislación española en aplicación de normas de Derecho Internacional público.

c. Cuando el responsable del tratamiento no esté establecido en territorio de la Unión Europea y utilice en el tratamiento de datos medios situados en territorio español, salvo que tales medios se utilicen únicamente con fines de tránsito.

2. El régimen de protección de los datos de carácter personal que se establece en la presente Ley Orgánica no será de aplicación:

a. A los ficheros mantenidos por personas físicas en el ejercicio de actividades

exclusivamente personales o domésticas.

b. A los ficheros sometidos a la normativa sobre protección de materias clasificadas.

c. A los ficheros establecidos para la investigación del terrorismo y de formas graves de delincuencia organizada. No obstante, en estos supuestos el responsable del fichero comunicará previamente la existencia del mismo, sus características generales y su finalidad a la Agencia de Protección de Datos.

3. Se regirán por sus disposiciones específicas, y por lo especialmente previsto, en su caso, por esta Ley Orgánica los siguientes tratamientos de datos personales:

a. Los ficheros regulados por la legislación de régimen electoral.

b. Los que sirvan a fines exclusivamente estadísticos, y estén amparados por la legislación estatal o autonómica sobre la función estadística pública.

c. Los que tengan por objeto el almacenamiento de los datos contenidos en los informes personales de calificación a que se refiere la legislación del régimen del personal de las Fuerzas Armadas.

d. Los derivados del Registro Civil y del Registro Central de penados y rebeldes.

e. Los procedentes de imágenes y sonidos obtenidos mediante la utilización de videocámaras por las Fuerzas y Cuerpos de Seguridad, de conformidad con la legislación sobre la materia.

Artículo 3. Definiciones.

A los efectos de la presente Ley Orgánica se entenderá por:

a. Datos de carácter personal: cualquier información concerniente a personas físicas identificadas o identificables.

b. Fichero: todo conjunto organizado de datos de carácter personal, cualquiera que fuere la forma o modalidad de su creación, almacenamiento, organización y acceso.

c. Tratamiento de datos: operaciones y procedimientos técnicos de carácter automatizado o no, que permitan la recogida, grabación, conservación, elaboración, modificación, bloqueo y cancelación, así como las cesiones de datos que resulten de comunicaciones, consultas, interconexiones y transferencias.

d. Responsable del fichero o tratamiento: persona física o jurídica, de naturaleza pública o privada, u órgano administrativo, que decida sobre la finalidad, contenido y uso del tratamiento.

e. Afectado o interesado: persona física titular de los datos que sean objeto del tratamiento a que se refiere el apartado c) del presente artículo.

f. Procedimiento de disociación: todo tratamiento de datos personales de modo que la información que se obtenga no pueda asociarse a persona identificada o identificable.

g. Encargado del tratamiento: la persona física o jurídica, autoridad pública, servicio o cualquier otro organismo que, sólo o conjuntamente con otros, trate datos personales por cuenta del responsable del tratamiento.

h. Consentimiento del interesado: toda manifestación de voluntad, libre, inequívoca, específica e informada, mediante la que el interesado consienta el tratamiento de datos personales que le conciernen.

i. Cesión o comunicación de datos: toda revelación de datos realizada a una persona distinta del interesado.

j. Fuentes accesibles al público: aquellos ficheros cuya consulta puede ser realizada, por cualquier persona, no impedida por una norma limitativa o sin más exigencia que, en su caso, el abono de una contraprestación. Tienen la consideración de fuentes de acceso público, exclusivamente, el censo promocional, los repertorios telefónicos en los términos previstos por su normativa específica y las listas de personas pertenecientes a grupos de profesionales que contengan únicamente los datos de nombre, título, profesión, actividad, grado académico, dirección e indicación de su pertenencia al grupo. Asimismo, tienen el carácter de fuentes de acceso público los diarios y boletines oficiales y los medios de comunicación.